

ARION DE CAMPOS JUNIOR

**O USO DE MÚLTIPLOS ENXAMES NA OTIMIZAÇÃO DE  
PROBLEMAS COM VÁRIOS OBJETIVOS**

CURITIBA

2014

ARION DE CAMPOS JUNIOR

**O USO DE MÚLTIPLOS ENXAMES NA OTIMIZAÇÃO DE  
PROBLEMAS COM VÁRIOS OBJETIVOS**

Tese apresentada ao Programa de Pós-Graduação em Informática do Setor de Ciências Exatas da Universidade Federal do Paraná, como requisito parcial à obtenção do título de Doutor em Ciência da Computação.

Orientadora: Prof.<sup>a</sup> Dr.<sup>a</sup> Aurora T. R. Pozo

Co-orientador: Prof. Dr. Elias P. Duarte Jr.

CURITIBA

2014

---

C198u

Campos Junior, Arion de

O uso de múltiplos enxames na otimização de problemas com vários objetivos / Arion de Campos Junior. – Curitiba, 2014.  
198f. : il. color. ; 30 cm.

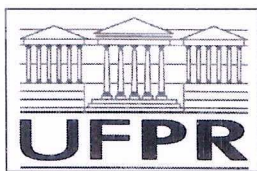
Tese (doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-graduação em Informática, 2014.

Orientadora: Aurora T. R. Pozo -- Co-orientador: Elias P. Duarte Jr..  
Bibliografia: p. 158-174.

1. Algoritmos de computador - Otimização. 2. Programação heurística. 3. Partículas I. Universidade Federal do Paraná. II. Pozo, Aurora T. R.. III. Duarte Jr., Elias P.. IV. Título.

CDD: 006.3824

---




Ministério da Educação  
Universidade Federal do Paraná  
Programa de Pós-Graduação em Informática


### PARECER


Nós, abaixo assinados, membros da Banca Examinadora da defesa do aluno de Doutorado em Ciência da Computação, Arion de Campos Junior, avaliamos a tese de doutorado intitulada “*O uso de múltiplos enxames na otimização de problemas com vários objetivos*”, cuja defesa pública foi realizada no dia 14 de fevereiro de 2014, às 13:30 horas, no Departamento de Informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após avaliação, decidimos pela:

☒ **aprovação** do candidato. ☐ **reprovação** do candidato.

Curitiba, 14 de fevereiro de 2014.

  
Prof. Dra. Aurora Trinidad Ramirez Pozo  
DINF/UFPR – Orientadora

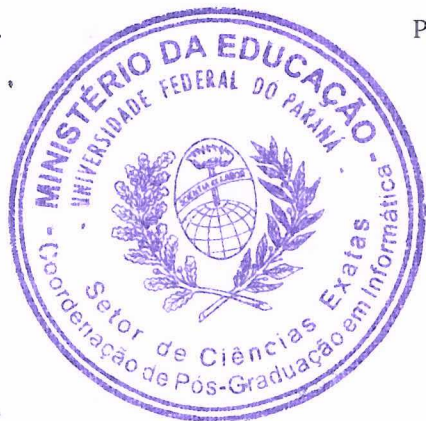
  
Prof. Dr. Elias Procópio Duarte Júnior  
DINF/UFPR – Coorientador

  
Prof. Dr. Andre Carlos Ponce de Leon Ferreira de Carvalho  
USP – Membro Externo

  
Profa. Dra. Myriam Regattieri De Biase da Silva Delgado  
UTFPR – Membro Externo

  
Prof. Dr. Leandro dos Santos Coelho  
UFPR – Membro Externo

  
Prof. Dr. Julio Cesar Nievola  
PUC/PR – Membro Externo



Dedico este trabalho à minha esposa, Juliana; à minha mãe, Lucia; ao meu pai, Arion e ao meu filho, Heitor.

## **AGRADECIMENTOS**

Inicialmente, gostaria de agradecer à professora Aurora Pozo. Além da firme e austera orientação recebida, sou grato pela confiança e amizade cultivadas ao longo dos anos. Agradeço ao professor Elias Duarte Jr., por sua valorosa co-orientação e pelo incentivo no decorrer deste trabalho.

Aos professores e funcionários da Universidade Estadual de Ponta Grossa e da Universidade Federal do Paraná, que possibilitaram o desenvolvimento deste trabalho.

À UEPG e à Fundação Araucária pelo financiamento deste doutorado.

Aos companheiros dos grupos de pesquisa da UFPR, em especial ao C-Bio/GrES, sempre solícitos em ajudar com valiosas sugestões e em proporcionar momentos de descontração.

À minha esposa, Juliana, meu amor, agradeço por seu companheirismo e apoio incondicional em todos esses anos.

Ao meu filho, Heitor. Sem dúvidas, a motivação vital para a realização deste trabalho.

Aos meus pais e à minha irmã, por todos esses anos de dedicação, suporte e amor.

## SUMÁRIO

<b>LISTA DE FIGURAS</b>	<b>vii</b>
<b>LISTA DE TABELAS</b>	<b>viii</b>
<b>LISTA DE SÍMBOLOS</b>	<b>xi</b>
<b>RESUMO</b>	<b>xii</b>
<b>ABSTRACT</b>	<b>xiii</b>
<b>1 INTRODUÇÃO</b>	<b>18</b>
1.1 Computação Evolutiva & PSO . . . . .	20
1.2 PSO com Múltiplos Enxames . . . . .	22
1.3 PSO para Multi e Vários Objetivos . . . . .	24
1.4 Contribuições . . . . .	25
1.5 Organização da tese . . . . .	27
<b>2 OTIMIZAÇÃO MULTIOBJETIVO E OTIMIZAÇÃO COM VÁRIOS OBJETIVOS BASEADA EM ALGORITMOS EVOLUTIVOS</b>	<b>29</b>
2.1 Otimização mono-objetivo . . . . .	30
2.2 Otimização multiobjetivo . . . . .	31
2.3 Métricas de desempenho para avaliar soluções de problemas com mais de um objetivo . . . . .	38
2.4 Otimização com vários objetivos . . . . .	43
2.5 Algoritmos evolutivos para resolver MOPs . . . . .	47
2.5.1 NSGA-II - <i>Nondominated Sorting Genetic Algorithm II</i> . . . . .	48
2.5.2 MOEA/D - <i>Multiobjective Evolutionary Algorithm Based on De- composition</i> . . . . .	50

<b>3</b>	<b>A OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS</b>	<b>55</b>
3.1	Otimização por enxame de partículas . . . . .	55
3.2	Modificações na fórmula de atualização da velocidade da partícula do algoritmo PSO . . . . .	57
3.2.1	Equação da velocidade com peso da inércia $\omega$ . . . . .	58
3.2.2	Equação da velocidade com fator de constrição . . . . .	58
3.3	Topologias de vizinhança . . . . .	59
3.4	PSO com múltiplos enxames . . . . .	63
3.4.1	<i>Multiple Swarm PSO</i> - MSPSO . . . . .	63
3.4.2	Taxonomia para identificar e classificar as estratégias de otimi- zação cooperativa . . . . .	67
3.4.3	Estratégias para a distribuição de enxames . . . . .	69
<b>4</b>	<b>A OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS APLICADA À RESOLU- ÇÃO DE PROBLEMAS COM MAIS DE UM OBJETIVO</b>	<b>76</b>
4.1	<i>Multiple objectives PSO</i> - MOPSO . . . . .	76
4.2	O desafio para determinar os líderes no MOPSO . . . . .	79
4.3	Gerenciamento de soluções não dominadas na otimização multiobjetivo	81
4.4	MOPSO - Estado da arte . . . . .	84
4.4.1	MOPSOs baseados em decomposição . . . . .	92
<b>5</b>	<b>ESTRATÉGIAS DISTRIBUÍDAS PARA A RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO UTILIZANDO O ALGORITMO PSO</b>	<b>99</b>
5.1	Uma abordagem assíncrona para a resolução de problemas mono-objetivo utilizando o algoritmo PSO . . . . .	99
5.2	Algoritmos baseados em múltiplos enxames distribuídos para resolver problemas com vários objetivos . . . . .	104
5.2.1	Estratégia distribuída baseada em conceitos de dominância para a resolução de MaOPs . . . . .	105



5.2.2	Estratégias distribuídas baseada em decomposição de problemas para a resolução de MaOPs . . . . .	109
6	AVALIAÇÃO EMPÍRICA DOS MÉTODOS PROPOSTOS	113
6.1	Metodologia para a validação dos métodos propostos . . . . .	113
6.1.1	Ambiente simulado . . . . .	114
6.1.2	Os benchmarks adotados . . . . .	115
6.1.3	Testes estatísticos . . . . .	116
6.2	A otimização mono-objetivo com múltiplos enxames em várias topologias	118
6.2.1	Simulação: metodologia & parâmetros . . . . .	118
6.2.2	Resultados . . . . .	121
6.2.3	Influência do número de enxames . . . . .	127
6.3	A otimização com múltiplos enxames e baseada em conceitos de dominância para resolver problemas com vários objetivos . . . . .	129
6.3.1	Simulação: metodologia & parâmetros . . . . .	130
6.3.2	Resultados . . . . .	131
6.4	A otimização com múltiplos enxames baseada em decomposição para resolver problemas com vários objetivos . . . . .	139
6.4.1	Simulação: metodologia & parâmetros . . . . .	140
6.4.2	Resultados . . . . .	142
7	CONCLUSÃO	156
	REFERÊNCIAS	158
A	TESTES ESTATÍSTICOS	175
A.1	Múltiplas comparações . . . . .	175
A	BENCHMARKS UTILIZADOS	193
A.1	Funções de benchmarks com um critério . . . . .	193
A.2	Funções de benchmarks com mais de um critério . . . . .	194

## LISTA DE FIGURAS

2.1	Fronteira de Pareto de um conjunto de soluções para um problema de minimização com dois objetivos. . . . .	33
2.2	Quatro tipos de fronteiras de Pareto para problemas de minimização. .	34
2.3	Exemplo de fronteira de Pareto (linha contínua) para um problema de minimização com destaque para o “joelho” da curva [Das, 1999]. . . . .	35
2.4	Exemplo de fronteira de Pareto (em vermelho) com os vetores objetivo ideal ( $z^*$ ), utopia ( $z^{**}$ ) e nadir ( $z^{nadir}$ ) [Branke et al., 2008]. . . . .	35
2.5	Exemplos com três conjuntos de soluções que ilustram relações entre convergência e diversidade. . . . .	37
2.6	Exemplo da métrica <i>hypervolume</i> para um problema de maximização com dois objetivos [Zitzler, 1999]. . . . .	39
2.7	Exemplo de fronteiras de Pareto real e obtida para a métrica GD para um problema com dois objetivos de minimização [Van Veldhuizen e Lamont, 2002]. . . . .	40
2.8	Exemplo de cálculo da métrica GDp: a linha representa a fronteira de Pareto real, enquanto que as setas correspondem à menor distância entre dois pontos entre as fronteiras conhecida e real. . . . .	42
2.9	Exemplo de cálculo da métrica IGDp: a linha representa a fronteira de Pareto real, enquanto que as setas correspondem à menor distância entre dois pontos das fronteiras real e conhecida. . . . .	43
2.10	Deterioração da habilidade de busca do algoritmo NSGA-II na resolução do problema DTLZ1 em relação ao aumento no número de objetivos. .	44
2.11	Visualização da fronteira de Pareto, segundo Obayashi e Sasaki [Obayashi e Sasaki, 2003]. . . . .	45
2.12	Procedimentos do NSGA-II [Deb et al., 2002a]. . . . .	50
3.1	Topologia de vizinhança em forma de grafo completo. . . . .	60

3.2	Topologia de vizinhança em forma de anel. . . . .	61
3.3	Topologia de vizinhança em forma de estrela. . . . .	61
3.4	Topologia de vizinhança em forma de árvore. . . . .	62
3.5	Topologia de vizinhança de forma arbitrária. . . . .	62
3.6	Duas abordagens para o compartilhamento de informações: (a) Modelo Co-evolutivo; (b) <i>Island Model</i> . . . . .	68
3.7	Exemplo de topologia dinâmica: inicialmente um grafo completo (a), que diminui sua conectividade à medida que o tempo passa (b) e (c), tornando-se uma topologia em anel (d). . . . .	72
3.8	Visualização de um hipercubo <i>3-cubo</i> . . . . .	73
3.9	Estratégias básicas de comunicação: remetentes representados por vértices em vermelho e destinatários representados por vértices em preto. . . . .	74
4.1	Ilustração do método PBI - <i>Penalty Boundary Intersection</i> [Zapotecas Martínez e Coello Coello, 2011]. . . . .	95
5.1	Topologias de enxames: <b>(a)</b> anel unidirecional; <b>(b)</b> anel bidirecional; <b>(c)</b> grafo completo; <b>(d)</b> hipercubo. . . . .	104
5.2	Compartilhamento de partículas: a seta indica a direção do compartilhamento de soluções entre os enxames remetente e destino(s). . . . .	107
6.1	Custo de comunicação para as Funções <i>Schwefel's 2.21</i> e <i>Rosenbrock</i> . . . . .	124
6.2	Custo de comunicação para a Função <i>Griewank</i> na topologia dinâmica. . . . .	125
6.3	Diagrama <i>Box-and-whisker</i> para a função de <i>Rosenbrock</i> : convergência e comunicação, respectivamente. . . . .	126
6.4	Boxplot dos resultados obtidos pela métrica IGDp para a função DTLZ4 que não apresentaram diferença estatística. . . . .	134
6.5	Padrão de comunicação para a função DTLZ2 (2 objetivos). . . . .	136
6.6	Padrão de comunicação para a função DTLZ2 (20 objetivos). . . . .	137
6.7	Padrão de comunicação para a função DTLZ4 (2 objetivos). . . . .	137
6.8	Padrão de comunicação para a função DTLZ4 (20 objetivos). . . . .	138

6.9	Boxplot dos desempenhos médios dos algoritmos para as métricas GDp, IGDp e <i>Spacing</i> . . . . .	139
6.10	Resultados médios relativos aos indicadores GDp e IGDp para o problema DTLZ1. . . . .	144
6.11	Resultados médios relativos aos indicadores GDp e <i>Spacing</i> para o problema DTLZ3. . . . .	147
6.12	Resultados médios relativos aos indicadores GDp e <i>Spacing</i> para o problema DTLZ4. . . . .	149
6.13	Resultados médios relativos aos indicadores IGDp e Hypervolume para o problema DTLZ5. . . . .	151
6.14	Resultados médios relativos aos indicadores IGDp e Hypervolume para o problema DTLZ7. . . . .	153
A.1	Fronteiras de Pareto real relativas ao conjunto de problemas DTLZ para três objetivos. . . . .	198

## LISTA DE TABELAS

6.1	Resultados para cada função considerando diferentes topologias. . . .	121
6.2	Custos de comunicação. . . . .	122
6.3	Ranqueamento das topologias baseado no <i>Success Performance</i> . . . .	127
6.4	Desempenho dos múltiplos enxames com 8, 16 e 32 subpopulações para as funções <i>Sphere</i> e <i>Rosenbrock</i> . . . . .	128
6.5	Indicadores de desempenho do problema DTLZ2. . . . .	132
6.6	Indicadores de desempenho do problema DTLZ4. . . . .	132
6.7	Configuração adotada para os algoritmos propostos (NA: Não se Aplica). 142	
6.8	Indicadores de desempenho do problema DTLZ1. . . . .	143
6.9	Indicadores de desempenho do problema DTLZ2. . . . .	145
6.10	Indicadores de desempenho do problema DTLZ3. . . . .	146
6.11	Indicadores de desempenho do problema DTLZ4. . . . .	148
6.12	Indicadores de desempenho do problema DTLZ5. . . . .	150
6.13	Indicadores de desempenho do problema DTLZ6. . . . .	152
6.14	Indicadores de desempenho do problema DTLZ7. . . . .	154

## LISTA DE SÍMBOLOS

<i>ACO</i>	<i>Ant Colony Optimization</i>
<i>AGA</i>	<i>Adaptive Grid Archiving</i>
<i>AIS</i>	<i>Artificial Immune Systems</i>
<i>AR</i>	<i>Average Ranking</i>
<i>BCO</i>	<i>Bee Colony Optimization</i>
<i>BI</i>	<i>Boundary Intersection</i>
<i>CCPSO</i>	<i>Cooperative Coevolutionary PSO</i>
<i>CD</i>	<i>Crowding Distance</i>
<i>CDAS</i>	<i>Controlling Dominance Area of Solutions</i>
<i>COMPSO</i>	<i>Cooperative Micro-Particle Swarm Optimization</i>
<i>CPSO</i>	<i>Cooperative Particle Swarm Optimizer</i>
<i>DE</i>	<i>Differential Evolution</i>
<i>DM</i>	<i>Decision Maker</i>
<i>dMOPSO</i>	<i>decomposition-Based MOPSO</i>
<i>DMS – PSO</i>	<i>Dynamic Multi-Swarm - PSO</i>
<i>DSMOPSO</i>	<i>Distance Sorting based Multi-Objective Particule Swarm Optimization</i>
<i>DTMPSO</i>	<i>Different Topology Multi-swarm PSO</i>
<i>EA</i>	<i>Evolutionary Algorithms</i>
<i>EC</i>	<i>Evolutionary Computation</i>

<i>EMO</i>	<i>Evolutionary Multi-Objective Optimization</i>
<i>GA</i>	<i>Genetic Algorithm</i>
<i>GD</i>	<i>Generational Distance</i>
<i>HV</i>	<i>Hipervolume</i>
<i>IB</i>	<i>Indicator Based</i>
<i>IGD</i>	<i>Inverted Generational Distance</i>
<i>MaOP</i>	<i>Many-Objective Optimization Problem</i>
<i>MGA</i>	<i>Multi-level Grid Archiving</i>
<i>MOEA</i>	<i>Multi-Objective Evolutionary Algorithm</i>
<i>MOEA/D</i>	<i>Multiobjective Evolutionary Algorithm based on Decomposition</i>
<i>MOP</i>	<i>Multi-Objective Optimization Problem</i>
<i>MOPSO</i>	<i>Multi-Objective Particle Swarm Optimization</i>
<i>MOPSO – PS</i>	<i>Multi-Objective Particle Swarm Optimization with Preference-based Sorting</i>
<i>MOPSO/D</i>	<i>Multi-Objective Particle Swarm Optimization based on Decomposition</i>
<i>MSPSO</i>	<i>Multi-Swarms PSO</i>
<i>NSGA</i>	<i>Non-dominated Sorting Genetic Algorithm</i>
<i>OR</i>	<i>Operational Research</i>
<i>PAES</i>	<i>Pareto Archived Evolution Strategy</i>
<i>PBI</i>	<i>Penalty Boundary Intersection</i>
<i>PO</i>	<i>Pesquisa Operacional</i>

*PSO*      *Particle Swarm Optimization*

*SD*        *Sistemas Distribuídos*

*SDMOPSO* *Smart Multi-Objective Particle Swarm Optimisation Using Decomposition*

*SMPL*    *SiMulation Programming Language*

*SOM*      *Self-Organising Maps*

*SOP*      *Single-objective Optimization Problem*

*SP*        *Success Performance*

*TD*        *Tomador de Decisão*

*WSA*      *Weighted Sum Approach*



## RESUMO

A computação bioinspirada permite a resolução de uma gama de problemas computacionais. Dentre as várias meta-heurísticas existentes, o algoritmo PSO (*Particle Swarm Optimization*) tem sido aplicado eficientemente para resolver problemas de otimização. Inicialmente empregado na resolução de problemas com um objetivo, a técnica tem sido investigada para solucionar problemas multiobjetivo. O principal objetivo desta tese de doutorado é a proposta de estratégias distribuídas para a execução do algoritmo PSO em diversas topologias conectando múltiplos enxames para resolver problemas com vários objetivos. A adoção de múltiplos enxames parte da constatação de que a otimização pode consumir onerosos recursos computacionais. Assim, investigar e propor novos métodos para a execução do algoritmo de forma paralela e distribuída torna-se uma iniciativa relevante. Neste trabalho, os indivíduos do algoritmo PSO são divididos em subpopulações independentes entre si e que ocasionalmente compartilham indivíduos. Diversas topologias e estratégias de comunicação para conectar os enxames foram investigadas, que determinam quais subpopulações trocam informações entre si. A influência exercida pela topologia na otimização de problemas com um objetivo é avaliada. Esta investigação inicial serviu para verificar se o uso de múltiplos enxames é relevante. Considerando os resultados obtidos, pôde-se constatar que esse modelo exerce um efeito positivo no processo de otimização. Foi possível indicar quais topologias apresentam melhor desempenho e qual a configuração, em termos de número de subpopulações, é mais eficiente. Tais constatações foram motivações para conceber estratégias distribuídas para resolver problemas com vários objetivos, incluindo uma estratégia baseada na decomposição de funções. Estudos empíricos são conduzidos para avaliar o impacto da otimização cooperativa, incluindo fatores relacionados à comunicação exigida entre as subpopulações. A partir desses resultados foi possível determinar qual estratégia baseada no algoritmo PSO é mais indicada, considerando as características de diferentes problemas de otimização.

## ABSTRACT

Bioinspired computation allows the resolution of a wide variety of computational problems. Among the many existing meta-heuristics, the PSO (Particle Swarm Optimization) algorithm has been effectively applied to solve several optimization problems. First employed for solving single objective problems, there are clearly advantages of using this technique to solve multiobjective problems. The main goal of this thesis is the proposal of distributed strategies for running PSO algorithm on several topologies connecting multiple swarms to solve problems with many objectives. The adoption of multiple swarms derives from the fact that the optimization can consume expensive computational resources. So, the investigation and proposal of new methods for the implementation of parallel and distributed algorithms becomes a relevant effort. The parallelization technique employed in this work consists of grouping PSO individuals in independent subpopulations, which occasionally share individuals. We have investigated the impact of several topologies and communication strategies that dictate how subpopulations exchange information among themselves. The influence of the topology single objective problems is first investigated. This initial research helped to determine whether the use of multiple swarms is relevant. Considering these results, it was noted that this model has a positive effect in the optimization process. It was possible to indicate which topologies are more efficient and the performance evaluation in terms of number of subpopulations. These findings were motivations for designing distributed strategies for solving problems with multiple objectives, including a strategy based on decomposition of functions. Empirical studies were conducted to assess the impact of cooperative optimization, including factors related to communication required between subpopulations. From these results it was possible to determine which optimization strategy based on PSO algorithm is more appropriate, considering the specific characteristics of different optimization problems.

## CAPÍTULO 1

### INTRODUÇÃO

A computação evolutiva (EC, do inglês *Evolutionary Computation*), inspirada no comportamento dos seres vivos, tem sido aplicada para a resolução de problemas de otimização em diversas áreas [Reyes-sierra e Coello, 2006, Coello, 1999, Hong-Xia et al., 2008, Luke, 2009]. Os métodos da EC procuram por uma “solução” entre um enorme número de possibilidades e não requerem qualquer conhecimento prévio do problema a ser resolvido. Várias meta-heurísticas, no contexto da EC, têm sido utilizadas com sucesso para resolver problemas de otimização [Guo et al., 2010, Dorigo et al., 1996, Dasgupta, 1997], inclusive a Otimização por Enxame de Partículas (PSO<sup>1</sup>, do inglês *Particle Swarm Optimization*) [Kennedy e Eberhart, 1995], apresentada por Kennedy e Eberhart em 1995. O PSO é uma meta-heurística de otimização estocástica que tem sido muito utilizada em diversas aplicações. Esse algoritmo é baseado na simulação do comportamento coletivo de animais (chamados partículas) procurando alimento ou fugindo de predadores. Cada partícula é considerada uma possível solução de um problema e a atualização de seus valores pode ser influenciada por outras partículas. Essa relação de influência depende de como as partículas estão organizadas no espaço de busca. Sucessivas atualizações conduzem a resultados que tendem à solução ótima.

Apesar de ser considerada uma técnica eficiente, o desempenho do PSO é degradado à medida que o tamanho do problema aumenta [Li et al., 2010]. O tamanho de um problema é medido em dimensões e quanto maior a dimensão do problema, maior o custo computacional para resolvê-lo. Diversas modificações no PSO original foram elaboradas para melhorar seu desempenho [Clerc, 1999, Shi e Eberhart, 1998, Parsopoulos, 2009, Angeline, 1998]. Neste trabalho destacamos a adoção de múltiplos

---

<sup>1</sup>Otimização baseada em partículas é uma técnica de inteligência coletiva, porém na área de otimização multiobjetivo é comum ser classificada como EC.

enxames [Li e Xiao, 2008]. Variações com múltiplos enxames têm apresentado resultados promissores em várias aplicações, inclusive para resolver problemas com mais de um objetivo. Nesta tese, são propostos métodos baseados em PSO com múltiplos enxames para resolver problemas de otimização. Experimentos são conduzidos para validar nossa estratégia considerando a otimização de problemas com um objetivo. O impacto da topologia na resolução de problemas complexos utilizando múltiplos enxames que compartilharam suas experiências assincronamente é investigado. Diferentes topologias e estratégias de comunicação são analisadas e o método distribuído proposto é utilizado para resolver funções de *benchmark* tradicionais. Confrontamos tais resultados aos obtidos por um modelo de otimização centralizado, validando nossa estratégia.

Na segunda contribuição desta tese, estratégias distribuídas baseadas no PSO com múltiplos e independentes enxames são desenvolvidas para resolver problemas com vários objetivos. Como técnica de paralelização, foi empregado o método de divisão dos indivíduos do algoritmo PSO em subpopulações independentes entre si. Foram estabelecidos métodos de comunicação assíncronos, desvinculados de intervalos de iterações fixos. Estes métodos permitiram o compartilhamento ocasional de indivíduos entre enxames distintos para ajudar a explorar regiões prominentes do espaço de busca. Finalmente, um método distribuído e baseado na técnica de decomposição para a resolução de problemas com vários objetivos é apresentado e validado. Nesta técnica, um problema multiobjetivo é separado em vários subproblemas mono-objetivos. Para cada subproblema, um conjunto de diferentes pesos é associado a cada objetivo original do problema multiobjetivo. Esse conjunto de pesos e seus respectivos subproblemas são distribuídos entre os múltiplos enxames. Isto permite que um valor de aptidão seja obtido para cada subproblema, considerando importâncias diversificadas para cada objetivo.

O restante deste capítulo está organizado da seguinte maneira. A Seção 1.1 apresenta as considerações iniciais sobre a utilização dos algoritmos evolutivos para resolver problemas, em especial a otimização por enxame de partículas. A Seção 1.2

apresenta as principais motivações que influenciaram a investigação do uso de múltiplos enxames distribuídos para resolver problemas de otimização. A Seção 1.3 aborda a utilização do PSO para resolver problemas com mais de um objetivo. A seguir, as principais contribuições deste trabalho são apresentadas na Seção 1.4 e, finalmente, a organização desta tese é exposta na Seção 1.5.

## 1.1 Computação Evolutiva & PSO

A solução de muitos problemas requer a busca através de um grande número de possibilidades, demandando um elevado custo computacional. Segundo Mitchell [Mitchell, 1998], alguns problemas precisam de soluções que sejam *adaptáveis* para continuar a ter bom desempenho caso o ambiente se altere; outros problemas requerem soluções que sejam *inovadoras* tendo em vista suas características originais; e, finalmente, muitos problemas computacionais exigem soluções *complexas* que são difíceis de serem obtidas manualmente. Nesse cenário, a computação evolutiva mostra-se adequada, pois consiste de métodos que procuram por uma “solução” entre um enorme número de possibilidades e não requerem qualquer conhecimento prévio do problema a ser resolvido.

Várias meta-heurísticas bioinspiradas têm sido utilizadas com sucesso para resolver problemas de otimização. Exemplos incluem o Algoritmo Genético (GA, do inglês *Genetic Algorithm*) [Guo et al., 2010], introduzido inicialmente por Holland nos anos 70. O GA é um algoritmo estocástico baseado nos princípios da seleção natural que tem sido aplicado satisfatoriamente em problemas de otimização e aprendizado de máquina. O algoritmo de Otimização de Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*), proposto por Dorigo [Dorigo et al., 1996] é um algoritmo probabilístico inspirado na observação do comportamento das formigas ao saírem para encontrar comida. Os Sistemas Imunológicos Artificiais (AIS, do inglês *Artificial Immune Systems*) [Dasgupta, 1997], baseados nos princípios do sistema imunológico dos vertebrados também são fonte de inspiração para o desenvolvimento de metodologias in-

teligentes para resolução de problemas. A Otimização por Colônia de Abelhas (BCO, do inglês *Bee Colony Optimization*), baseada nos princípios básicos da inteligência coletiva das abelhas [Teodorovic e Dell'orco, 2005]. A Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) [Kennedy e Eberhart, 1995], apresentada por Kennedy e Eberhart em 1995 é um algoritmo inspirado na observação do movimento coletivo de animais e insetos na busca por alimento. O algoritmo realiza a otimização de um problema tentando melhorar a solução iterativamente. Esse algoritmo é alvo desse trabalho.

O PSO é uma meta-heurística de otimização estocástica que tem sido muito utilizada em diversas aplicações. Esse algoritmo é baseado na simulação do comportamento social observado em animais e insetos e é considerada uma técnica simples - pois depende de poucos argumentos - e eficiente [Chakraborty et al., 2010]. Cada partícula é considerada uma possível solução de um problema e a atualização de seus valores pode ser influenciada por outras partículas. Essa relação de influência depende de como as partículas estão organizadas no espaço de busca. Através de frequentes atualizações, os resultados aproximam-se à solução ótima. Funções de aptidão (*fitness*) são empregadas para avaliar a qualidade de um indivíduo.

Segundo Padhye *et al.* [Padhye et al., 2009], PSO é uma técnica baseada em população, similar aos algoritmos genéticos em muitos aspectos, exceto que potenciais soluções movem-se ao invés de evoluírem no espaço de busca. Em comparação com os algoritmos evolutivos, anteriormente mencionados, é considerada uma técnica que encontra bons resultados rapidamente. Entretanto, essa característica apresenta um efeito negativo em certos problemas, pois a rápida convergência pode induzir os resultados para falsos ótimos globais (ótimos locais). Diversas modificações no PSO original foram elaboradas para melhorar seu desempenho e para contornar suas deficiências. Essas variações incluem alterações na fórmula de atualização da velocidade da partícula [Clerc, 1999, Shi e Eberhart, 1998], divisão do espaço de busca [Parasopoulos, 2009], incorporação de conceitos de algoritmos genéticos [Angeline, 1998], além da adoção de múltiplos enxames [Li e Xiao, 2008], que é o foco da investigação

realizada neste trabalho.

## 1.2 PSO com Múltiplos Enxames

Uma das principais características da otimização por enxame de partículas é sua simplicidade. É uma técnica que pode ser facilmente adaptada e modificada. Entre diversas variações, o PSO com múltiplos e cooperativos enxames para resolver funções de otimização revela-se uma estratégia promissora.

Quando se pondera o desempenho do algoritmo PSO, um aspecto de fundamental importância diz respeito à maneira como os indivíduos de um enxame trocam as informações entre si. Essa troca é condicionada à forma como os indivíduos estão conectados e é determinada por sua vizinhança (também conhecida como rede social). Assim, a relação de influência entre os indivíduos é dependente da topologia de vizinhança (também chamada de *estrutura social*) implementada. Segundo Kennedy [Kennedy, 1999], quanto maior a quantidade de vizinhos de uma partícula, mais rápida será sua convergência, porém, existe uma chance maior do resultado ficar estagnado em ótimos locais. Neste contexto, a rede social está associada à relação entre indivíduos pertencentes a um mesmo enxame. Entretanto, quando são consideradas as topologias de vizinhança entre subpopulações, ainda existem lacunas a serem preenchidas. A topologia entre diferentes enxames apresenta novas e encorajadoras questões a serem consideradas.

Aspectos associados à conectividade entre subpopulações e às relações de compartilhamento de experiências entre as mesmas também exigem atenção. A adoção de múltiplos enxames parte da constatação que a otimização por enxame de partículas, assim como qualquer outra meta-heurística baseada em população, pode consumir onerosos recursos computacionais. Esse custo é referente ao elevado número de avaliações que são realizadas em função da quantidade de indivíduos e iterações despendidas pelo processo de otimização. Consequentemente, métodos paralelos tornam-se atraentes [Parra et al., 2009].

Existem várias maneiras de se obter paralelismo nas atividades de otimização [Luke, 2009]. Exemplos incluem realizar várias execuções simultaneamente em paralelo e aproveitar a característica multitarefa dos sistemas operacionais para executar múltiplas linhas de otimização (múltiplas *threads*) em paralelo em uma mesma máquina. Também é possível utilizar *hardware* específico para processamento matemático de alto desempenho, como computadores vetoriais ou executar a otimização baseada no modelo Mestre/Escravo (*Master-Slave*). Em um exemplo baseado neste modelo, uma máquina pode enviar indivíduos para que máquinas remotas avaliem a aptidão e devolvam resultados.

A técnica de paralelização adotada neste trabalho consiste em permitir que subpopulações realizem execuções em paralelo e eventualmente troquem experiências entre si. Geralmente, essa experiência compreende indivíduos que possuem bons valores de aptidão (*fitness*). Esses métodos são chamados de *island models* (modelos ilha) [Luke, 2009]. Tal escolha se justifica pelo fato de que, além da melhora no desempenho (*speedup*) obtida pelas execuções em paralelo, esse modelo exerce um efeito positivo no processo de otimização como um todo. Segundo Luke [Luke, 2009], executar várias funções de aptidão em muitas populações é melhor do que executar uma única função de aptidão em uma única população. Vários bons indivíduos aumentam a possibilidade de explorar mais eficientemente o espaço de busca.

O modelo ilha consiste de um grupo de processos evolutivos que ocasionalmente trocam indivíduos entre si para ajudar a explorar áreas recém descobertas e potencialmente promissoras do espaço de busca [Luke, 2009]. Os critérios de escolha dos indivíduos variam entre as implementações, mas geralmente envolvem os de melhor aptidão. A escolha aleatória pode ser adotada, todavia é uma alternativa que explora menos eficientemente o espaço de busca. O objetivo principal do modelo ilha é utilizar de forma eficiente os recursos computacionais, geralmente utilizando um conjunto de computadores. Porém, existem outras alternativas, como dividir a população em subpopulações - algumas vezes chamadas de *demes* - todo o sistema tende a ficar mais diversificado, aumentando a capacidade de exploração; se a função de aptidão puder



ser dividida em diferentes componentes, o modelo ilha pode, teoricamente, tratar diferentes componentes em populações separadas.

Para configurar o modelo ilha é necessário definir a topologia entre as ilhas, que estabelece quais subpopulações trocarão informações entre si. Neste trabalho consideramos a adoção de diversas topologias para estabelecer a relação entre as ilhas. No modelo ilha, aspectos de comunicação devem ser considerados e um conjunto de questões torna-se pertinente:

- Quando uma ilha se comunica, o processamento das outras ilhas deve esperar ou não essa comunicação ser concluída?
- Qual é a capacidade de comunicação entre as ilhas (envolvendo tipo e quantidade de conexões)?
- Quais elementos de processamento estão conectados?
- Com que frequência a comunicação ocorre?
- Quantos indivíduos são enviados?

Todas essas questões devem ser consideradas ao definir os mecanismos de compartilhamento das informações, além dos muitos aspectos relacionados à topologia das subpopulações. Aqui cabe uma ressalva: as topologias investigadas nesse trabalho compreendem topologias entre as subpopulações. Ou seja, analisamos as topologias entre os enxames. Dentro de cada enxame, todos os indivíduos (partículas) são influenciados uns pelos outros.

### **1.3 PSO para Multi e Vários Objetivos**

A possibilidade de modificar o PSO para executar em múltiplos enxames permitiu o desenvolvimento de uma nova classe de algoritmos. Porém, mesmo considerando esta ramificação, ainda existem lacunas associadas à otimização distribuída quando as funções consideradas envolvem vários objetivos. O campo responsável pelo estudo

dos problemas com vários objetivos é chamado de Otimização com Vários Objetivos, do inglês *Many-Objective Optimization*.

Nesse contexto, problemas que necessitam otimizar mais de uma função simultaneamente são chamados de Problemas de Otimização Multiobjetivo (MOP, do inglês *Multi-objective Optimization Problems*) e são resolvidos por vários Algoritmos Evolutivos Multiobjetivo (MOEA, do inglês *Multi-Objective Evolutionary Algorithms*). Todavia, alguns problemas surgem quando muitos objetivos são tratados, especialmente mais que três. O principal obstáculo enfrentado pelos MOEAs nesta situação é a deterioração da habilidade de busca, porque quase todas as soluções geradas são soluções não dominadas. É difícil estabelecer uma preferência entre as soluções, consequentemente a escolha de soluções que apresentem diversidade e convergência não é trivial [Ishibuchi et al., 2009]. Assim, não há um único algoritmo que resolva de maneira definitiva essa situação.

Recentemente, alguns algoritmos evolutivos baseados em decomposição [Zhang e Li, 2007] foram propostos para solucionar problemas multiobjetivos. Versões baseadas em PSO foram utilizadas e apresentaram bons resultados. Apesar disso, o desenvolvimento de algoritmos PSO baseados em decomposição ainda é incipiente. Essa asserção justifica-se pelo fato de que há muito a ser explorado na otimização baseada em decomposição de funções, incluindo problemas com vários objetivos e o efeito da otimização distribuída em subpopulações.

Diante do exposto, podemos destacar que o uso de múltiplos enxames é uma abordagem que demanda estudos no contexto da otimização com vários objetivos. Investigações envolvendo MOPSOs fundamentados em conceitos tradicionais de dominância de Pareto e conceitos baseados em decomposição merecem ser abordadas.

## 1.4 Contribuições

As principais contribuições deste trabalho envolvem a apresentação de novas estratégias distribuídas baseadas em múltiplos enxames para a execução do algoritmo PSO

para resolver problemas de otimização. Tais estratégias conduziram às contribuições sumarizadas a seguir:

1. *Avaliação de diversas topologias de interconexão empregadas para definir a rede social entre múltiplos enxames do algoritmo PSO.* Várias topologias de vizinhança entre subpopulações no algoritmo PSO foram adaptadas. A influência no desempenho de problemas com características distintas foi avaliada. Neste cenário, concebemos estratégias de comunicação assíncronas que exerceram um efeito positivo no processo de otimização e que permitiram a exploração mais eficiente do espaço de busca. Propostas distribuídas com enxames executando o processo de otimização independentemente foram validadas. Uma série de funções de *benchmarks* foram testadas e resultados foram obtidos através de execuções distribuídas e referenciais. A variação da conectividade entre as subpopulações e sua influência no processo de otimização foram investigadas. O impacto exercido pelos diferentes graus de conectividade em diferentes funções foi investigado, inclusive através da implementação de mecanismos que dinamicamente modificaram a cooperação entre os enxames. Tais funções foram individualmente analisadas para aferir como graus maiores ou menores de conectividade impactaram o desempenho.
2. *Investigação do algoritmo PSO cooperativo para resolver problemas multi e com vários objetivos.* Realizamos a execução simultânea do processo de otimização em enxames independentes e consideramos novas e desafiadoras perspectivas, incluindo: a comunicação envolvida, a sincronização entre enxames, a manutenção de soluções não dominadas, o número de enxames, entre outros. A escalabilidade do PSO distribuído foi avaliada. Resultados foram extraídos através da execução de múltiplos enxames de forma cooperativa em diversos problemas de *benchmarks*. Tais resultados foram comparados com versões centralizadas para avaliar a escalabilidade dos algoritmos quando o número de objetivos é gradativamente aumentado. Os algoritmos foram avaliados considerando

métricas de desempenho para mensurar adequadamente a diversidade e convergências das soluções obtidas. Uma estratégia de PSO distribuído baseada em decomposição para otimizar problemas com vários objetivos foi proposta. Este algoritmo foi executado para solucionar todos os *benchmarks* da família de problemas multiobjetivos DTLZ [Deb et al., 2002b]. Experimentos foram conduzidos e análises empíricas baseadas em testes estatísticos foram realizadas. Os resultados obtidos foram comparados aos resultados de outros algoritmos, incluindo versões centralizadas e versões de PSO distribuído não baseados em decomposição.

## 1.5 Organização da tese

O restante desta tese está organizada conforme segue.

**Capítulo 2 - Otimização Multiobjetivo e Otimização com Vários Objetivos Baseada em Algoritmos Evolutivos:** este Capítulo aborda os principais conceitos relacionados à otimização de problemas. Inclui as otimizações mono-objetivo, multiobjetivo e com vários objetivos. Os principais indicadores de desempenho são expostos, assim como os MOEAs básicos que são necessários ao desenvolvimento desta tese.

**Capítulo 3 - A Otimização por Enxame de Partículas:** apresenta os conceitos da otimização por enxame de partículas. O algoritmo PSO original é apresentado e versões alternativas relevantes disponíveis na literatura relativas a esse algoritmo são consideradas. A definição de vizinhança é apresentada e aspectos associados à sua influência no processo de otimização são discutidos. Também aborda a utilização de múltiplos enxames na execução do algoritmo PSO, incluindo os aspectos de comunicação.

**Capítulo 4 - A Otimização por Enxame de Partículas Aplicada à Resolução de Problemas Com Mais de Um Objetivo:** este Capítulo considera a literatura disponível e trata dos aspectos associados à otimização por enxame de partículas para resolver problemas multiobjetivos. Os conceitos de melhores partículas locais e globais são

adotados no contexto multiobjetivo. As técnicas para seleção de líderes e arquivamento são tratadas e os principais algoritmos propostos para resolver problemas multiobjetivos usando o PSO são discutidos. Finalizando, este Capítulo contempla as principais estratégias baseadas em decomposição e que utilizam o PSO para resolver problemas com mais de um objetivo.

**Capítulo 5 - Estratégias Distribuídas para a Resolução de Problemas de Otimização Utilizando o Algoritmo PSO:** nesse Capítulo propomos abordagens cooperativas baseadas no algoritmo PSO para a resolução de problemas de otimização. O Capítulo contém duas Seções principais que tratam: i) uma estratégia distribuída para resolução de problemas mono-objetivo; ii) duas estratégias distribuídas para a resolução de problemas com vários objetivos. Com relação às duas estratégias: a primeira utiliza os conceitos de dominância de Pareto tradicionais e a segunda compreende um algoritmo distribuído para a resolução de problemas com vários objetivos baseado em decomposição de problemas.

**Capítulo 6 - Avaliação Empírica dos Métodos Propostos:** Neste Capítulo validamos nossas estratégias distribuídas para a resolução de problemas de otimização utilizando o algoritmo PSO. Esta validação é realizada através da avaliação de execuções distribuídas para resolver funções de *benchmarks* que são comparadas às execuções centralizadas. Os resultados experimentais da avaliação do desempenho da otimização com múltiplos exames distribuídos são descritos e discutidos.

**Capítulo 7 - Conclusão:** finalmente, as conclusões são aqui expostas, incluindo caminhos para futuros trabalhos.

## CAPÍTULO 2

### OTIMIZAÇÃO MULTIOBJETIVO E OTIMIZAÇÃO COM VÁRIOS OBJETIVOS BASEADA EM ALGORITMOS EVOLUTIVOS

Otimizar é a tarefa de encontrar uma ou mais soluções que correspondam à minimização ou maximização de um ou mais objetivos específicos e que satisfaçam todas as restrições que houver [Mitchell, 1998].

Problemas com mais de um objetivo surgem naturalmente e são comuns em áreas como economia, finanças, engenharia, matemática e várias outras e têm sido um desafio aos pesquisadores ao longo do tempo. Apesar da variedade de técnicas desenvolvidas na Pesquisa Operacional (OR, do inglês *Operation Research*) e outras técnicas para enfrentar esses problemas, a dificuldade envolvida, muitas vezes exige abordagens alternativas [Coello et al., 2007].

Algoritmos evolutivos (EA, do inglês *Evolutionary Algorithm*) são algoritmos estocásticos baseados em populações cujos indivíduos (soluções) participam de um processo em que evoluem iterativamente para novas soluções. Ao longo de quase duas décadas, tais algoritmos têm sido utilizados para resolver problemas de distintas áreas. Consequentemente, tornaram-se uma área relevante e com diversas aplicações. Segundo [Deb, 2011], a popularidade dos EAs se deve a muitos fatores, entre eles:

- não há necessidade de informações prévias sobre as funções a serem otimizadas;
- relativamente, são de simples implementação;
- podem ser adaptados e possuem uma aplicabilidade difundida.

De modo a compreender a otimização com muitos objetivos, inicialmente, a otimização mono-objetivo será abordada.

## 2.1 Otimização mono-objetivo

Um problema de otimização de objetivo único ou mono-objetivo envolve uma única função objetivo e, geralmente, tem como resultado uma única solução, chamada de solução ótima ou ideal.

Um problema de otimização mono-objetivo pode ser apresentado como na Definição 1 [Coello et al., 2007] e pode ser resolvido por muitas técnicas, incluindo diferentes algoritmos evolutivos.

**Definição 1. (Problema de Otimização Mono-objetivo Geral):** *Um problema de otimização mono-objetivo geral é definido como minimizar  $f(x)$  sujeito a um conjunto de restrições de desigualdade  $g_i(x) \leq 0$ ,  $i = \{1, \dots, m\}$ , e igualdade  $h_j(x) = 0$ ,  $j = \{1, \dots, p\}$ , com  $x$  restrito a um conjunto definido por  $\Omega$ . Uma solução minimiza a função escalar  $f(x)$  onde  $x$  é um vetor  $n$ -dimensional de variáveis de decisão  $x = (x_1, \dots, x_n)$  a partir de um conjunto  $x \in \Omega$ .*

Observar que  $g_i(x) \leq 0$  e  $h_j(x) = 0$  representam restrições que devem ser satisfeitas na otimização de  $f(x)$ .  $\Omega$  contém todos os valores de  $x$  possíveis que podem ser usados para satisfazer uma avaliação de  $f(x)$  e suas restrições. Também,  $x$  pode ser um vetor de variáveis discretas ou contínuas assim como  $f$ .

O método para encontrar o ótimo global (pode não ser único) de qualquer função é conhecido como **Otimização Global**. Em geral, o mínimo global de um problema mono-objetivo é apresentado na Definição 2 [Coello et al., 2007].

**Definição 2. (Mínimo Global de um Problema Mono-objetivo):** *Dada uma função  $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $\Omega \neq \emptyset$ , para  $x \in \Omega$  o valor  $f^* \triangleq f(x^*) > -\infty$  é chamado de **mínimo global** se, e somente se*

$$\forall x \in \Omega : f(x^*) \leq f(x). \quad (2.1)$$

*Por definição,  $x^*$  é a solução global mínima,  $f$  é a função objetivo, e o conjunto  $\Omega$  é a região factível de  $x$ . A meta em determinar a solução mínima global é chamada de*

**problema de otimização global** para um problema mono-objetivo.

Enquanto que na otimização de um problema mono-objetivo possa haver uma única solução ótima, isto é, não existe no conjunto considerado outro  $x$  com valor  $f(x)$  menor, os problemas multiobjetivo apresentam (como regra geral) um conjunto com incontáveis soluções.

## 2.2 Otimização multiobjetivo

Segundo [Coello et al., 2007], os problemas de otimização multiobjetivo, (também chamado otimização multicritério, multiperformance ou problema de otimização de vetor) podem ser definidos como um “*problema em determinar um vetor de variáveis de decisão que satisfaz restrições e otimiza uma função vetorial, cujos elementos representam as funções objetivo. Tais funções formam uma descrição matemática de critérios de desempenho que geralmente estão em conflito uns com os outros. Assim, o termo ‘otimizar’ significa encontrar uma solução desse tipo em que os valores de todas as funções objetivo sejam aceitáveis para o tomador de decisão (TD ou DM, do inglês Decision Maker)*”.

A solução dos Problemas de Otimização Multiobjetivo (MOP, do inglês *Multi-objective Optimization Problems*) é considerada uma importante área de pesquisa, pois além de serem problemas associados aos problemas do mundo real, existem muitas questões em aberto nessa área [Coello, 1999]. Nesses problemas, os objetivos a serem otimizados geralmente são conflitantes entre si, isto é, não existe uma única solução para estes problemas. A meta é encontrar um bom conjunto de soluções equilibradas, ou seja, um conjunto de soluções com um bom “*trade-off*” (compromisso) que representem as melhores possibilidades entre os objetivos [Reyes-sierra e Coello, 2006].

Matematicamente, o problema de otimização multiobjetivo pode ser escrito como segue:

Minimizar:

$$\vec{f} := [f_1(\vec{x}), f_2(\vec{x}), \dots, f_n(\vec{x})] \quad (2.2)$$



Sujeito a:

$$g_i(\vec{x}) \leq 0 \quad i = 1, 2, \dots, m \quad (2.3)$$

$$h_j(\vec{x}) = 0 \quad j = 1, 2, \dots, p \quad (2.4)$$

Onde  $\vec{x} = [x_1, x_2, \dots, x_n]^T$  é o vetor das variáveis de decisão;  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1 \dots n$  são as funções objetivo e  $g_i, h_j : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m, j = 1, \dots, p$  são as funções de restrição do problema.

Para descrever o conceito de otimização multiobjetivo, algumas definições serão apresentadas [Reyes-sierra e Coello, 2006, Coello et al., 2007]:

**Definição 3.** Dados dois vetores  $\vec{x}, \vec{y} \in \mathbb{R}^n$ , dizemos que  $\vec{x} \leq \vec{y}$  se  $x_i \leq y_i$  para  $i = 1, \dots, n$  e que  $\vec{x}$  **domina**  $\vec{y}$ , (denotado por  $\vec{x} \prec \vec{y}$ ) se  $\vec{x} \leq \vec{y}$  e  $\vec{x} \neq \vec{y}$ .

**Definição 4.** Dizemos que um vetor de variáveis de decisão  $\vec{x} \in X \subset \mathbb{R}^n$  é **não dominado** em relação a  $X$ , se não existe outro  $\vec{x}' \in X$  tal que  $f(\vec{x}') \prec f(\vec{x})$ .

**Definição 5.** Dada uma função  $F(\vec{x})$ , dizemos que um vetor de variáveis de decisão  $\vec{x} \in \Omega \subset \mathbb{R}^n$  ( $\Omega$  é uma região factível - feasible region) é uma solução **Pareto ótima** se ele é não dominado em relação a  $F$ .

**Definição 6.** O conjunto Pareto ótimo  $P^*$  é definido por  $P^* = \{\vec{x} \in \Omega \mid \vec{x} \text{ é Pareto ótimo}\}$ .

**Definição 7.** A Fronteira (ou Frente) de Pareto  $FP^*$  é definida por:  $FP^* = \{f(\vec{x}) \in \mathbb{R}^n \mid \vec{x} \in P^*\}$ .

A Figura 2.1 ilustra um caso de Fronteira de Pareto considerando duas funções objetivo. Assim, deseja-se determinar o conjunto Pareto ótimo de um conjunto  $F$  de todos os vetores de variáveis de decisão que satisfaçam 2.3 e 2.4. Nesse texto, quando uma solução não dominada é referenciada, significa dizer que seu vetor objetivo é não-dominado.

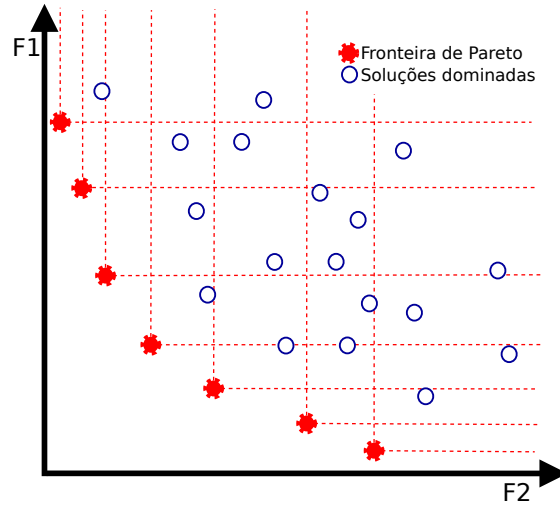


Figura 2.1: Fronteira de Pareto de um conjunto de soluções para um problema de minimização com dois objetivos.

De acordo com [Luke, 2009], as fronteiras de Pareto podem ser de vários tipos, como ilustrado pela Figura 2.2. Fronteiras convexas são curvadas para fora em direção às melhores soluções (Figura 2.2(b)). Fronteiras côncavas são curvadas para dentro (Figura 2.2(c)). Fronteiras não-convexas (Figura 2.2(a)) não são totalmente convexas, e elas incluem fronteiras côncavas como uma subcategoria. As fronteiras podem também ser descontínuas (Figura 2.2(d)), significando que existem regiões ao longo da fronteira que são simplesmente impossíveis de serem alcançadas: uma solução seria dominada por uma outra solução em outra parte de uma região factível ao longo da fronteira.

**Definição 8.** O **vetor Ideal** é o vetor que contém os melhores valores possíveis para cada função objetivo. O vetor ideal é denotado por:

$$F(\vec{y}) = [f_1^*(\vec{x}), f_2^*(\vec{x}), \dots, f_m^*(\vec{x})]$$

onde  $f_i^*(\vec{x})$  é o valor ótimo da função objetivo  $f_i(\vec{x})$ . Em geral  $f^*(\vec{y})$  é utilizado para definir a melhor posição possível no espaço de objetivos. Porém, em geral, essa posição não pode ser alcançada.

A seguir são apresentados alguns conceitos adicionais que são utilizados por diversos algoritmos de otimização multiobjetivo. Entre estes conceitos, destacam-se o **joelho da fronteira de Pareto**, o vetor objetivo **utopia** e o vetor objetivo **nadir**.

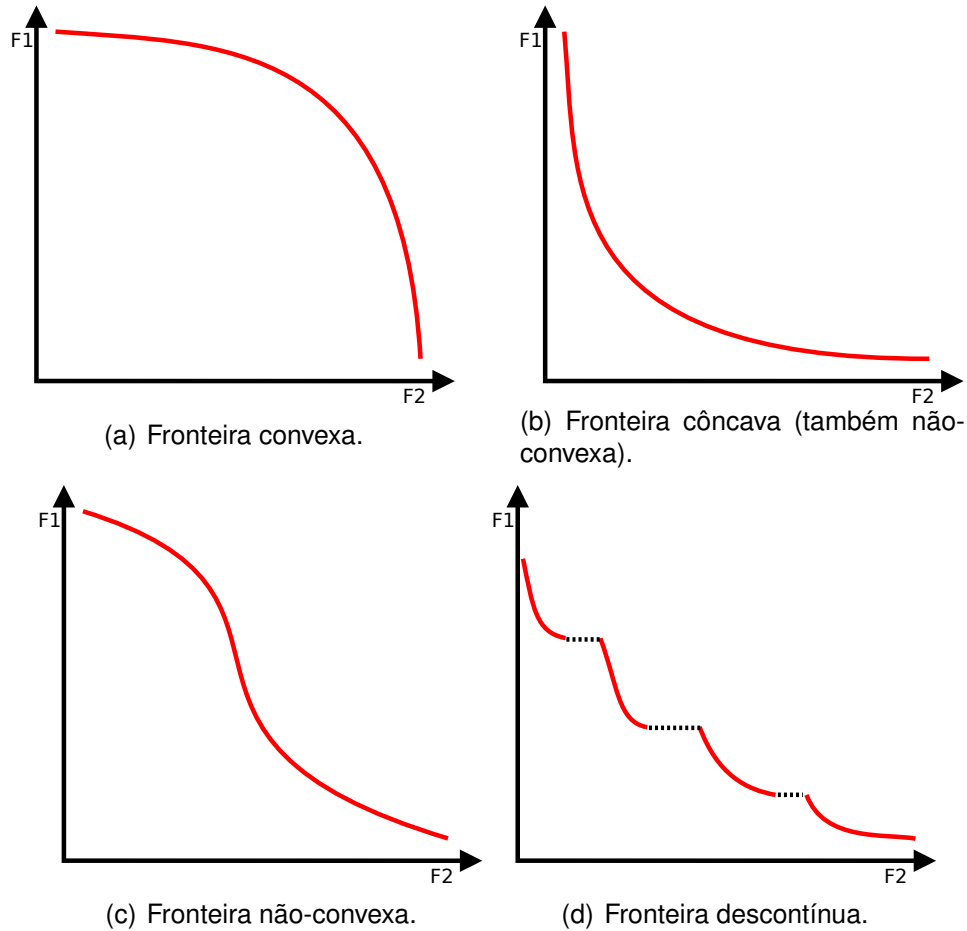


Figura 2.2: Quatro tipos de fronteiras de Pareto para problemas de minimização.

Das [Das, 1999] observou que, a partir da experiência prática, dado um conjunto de soluções que compõem a curva ou superfície de um dado problema multiobjetivo, o usuário geralmente opta por uma solução localizada “no meio” da curva. Esta solução compreende o ponto de maior protuberância da curva ou o “*joelho*” da curva. A Figura 2.3 ilustra uma fronteira de Pareto de um problema com dois objetivos de minimização onde o *joelho* da curva é destacado. O conceito de joelho é explorado por vários algoritmos de otimização [Das, 1999].

Como dito anteriormente, o vetor ideal é composto pelas soluções ótimas considerando cada objetivo, separadamente. Para um problema multiobjetivo de minimização, segundo [Branke et al., 2008], os componentes  $z_i^*$  do vetor ideal  $z^*$  são obtidos minimizando cada uma das funções objetivo individualmente, condicionadas à região factível. Um vetor estritamente melhor que  $z^*$  é chamado de **vetor objetivo utópico**  $z^{**}$ . Ou seja,  $z_i^{**} = z_i^* - \epsilon$  para  $i = 1, \dots, k$ , onde  $\epsilon$  é um pequeno escalar positivo.

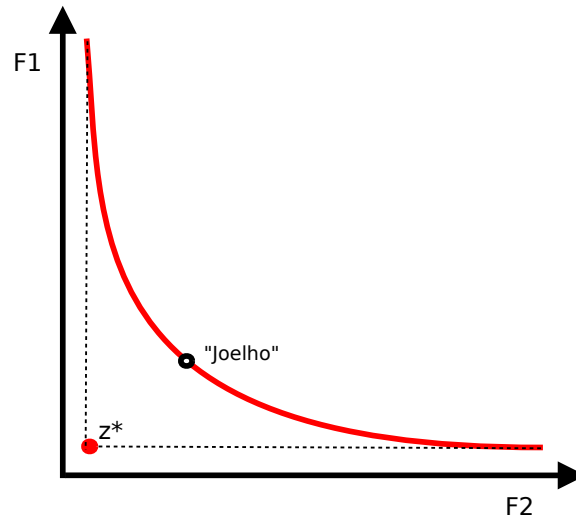


Figura 2.3: Exemplo de fronteira de Pareto (linha contínua) para um problema de minimização com destaque para o “joelho” da curva [Das, 1999].

Ainda de acordo com [Branke et al., 2008], para o mesmo problema de minimização, o vetor composto pelos limites superiores para cada objetivo considerando toda a fronteira de Pareto (não condicionado à região factível) é chamado de **vetor nadir**  $z^{nadir}$ . Ao contrário do vetor ideal  $z^*$ , em que cada função objetivo é condicionada à toda região factível, o vetor nadir  $z^{nadir}$  não é dependente desta região e geralmente é difícil de se obter. A Figura 2.4 ilustra os vetores ideal, utopia e nadir em um exemplo de problema com dois objetivos de minimização.

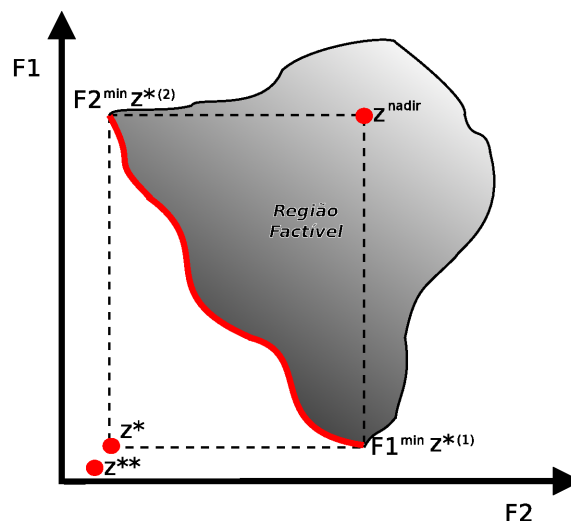


Figura 2.4: Exemplo de fronteira de Pareto (em vermelho) com os vetores objetivo ideal ( $z^*$ ), utopia ( $z^{**}$ ) e nadir ( $z^{nadir}$ ) [Branke et al., 2008].

A otimização multiobjetivo tem como meta encontrar um conjunto de soluções que

não são dominadas por nenhuma outra no espaço de objetivos e, na maioria das aplicações, a procura pelo conjunto Pareto ótimo é considerada um problema NP-Difícil [Coello et al., 2007]. Algoritmos evolutivos são boas opções para resolver problemas multiobjetivo porque lidam simultaneamente com um conjunto de possíveis soluções (chamada de população). Assim, pode-se encontrar um conjunto completo de soluções Pareto ótimo em uma única rodada do algoritmo. Ao invés de ter que executar um conjunto de rodadas independentes como é o caso das técnicas de programação matemática tradicionais [Luke, 2009].

De acordo com [Deb, 2011, Branke et al., 2008], estes algoritmos devem gerar um conjunto de soluções que deve possuir (i) uma boa convergência para a fronteira de Pareto ótima e (ii) uma boa diversidade nas soluções obtidas.

A convergência de um algoritmo diz respeito à constante aproximação, ao longo do tempo, do conjunto de soluções em direção à fronteira de Pareto real. Tal aproximação ocorre durante o processo iterativo do MOEA e é observada à medida que as soluções evoluem no espaço de busca das funções objetivo. Outra importante propriedade aspirada pelos MOEAs diz respeito à localização do conjunto de soluções no espaço de busca das funções. É desejável que o conjunto de soluções obtido esteja espalhado por diferentes regiões ao longo da fronteira de Pareto.

A Figura 2.5 ilustra três conjuntos de soluções distintos para um problema de otimização com dois objetivos de minimização. Em todos os casos, a curva contínua representa a fronteira de Pareto real e os círculos representam as respectivas soluções de cada conjunto de soluções. O conjunto ilustrado pela Figura 2.5(a) mostra algumas soluções com boa convergência próximas à fronteira de Pareto. Porém tais soluções estão concentradas em uma única região, não contendo soluções que cubram outras regiões da fronteira. Situação oposta é ilustrada pela Figura 2.5(b), em que observa-se um conjunto de soluções com boa diversidade, isto é, cobrindo todas as regiões ao longo da fronteira. Todavia, essas soluções apresentam uma convergência ruim, pois estão longe da fronteira de Pareto. Finalmente, uma situação em que o conjunto de soluções apresenta boa convergência e boa diversidade é ilustrada pela Figura 2.5(c).

As soluções apresentam-se próximas à fronteira de Pareto e estão equilibradamente bem distribuídas ao longo da curva.

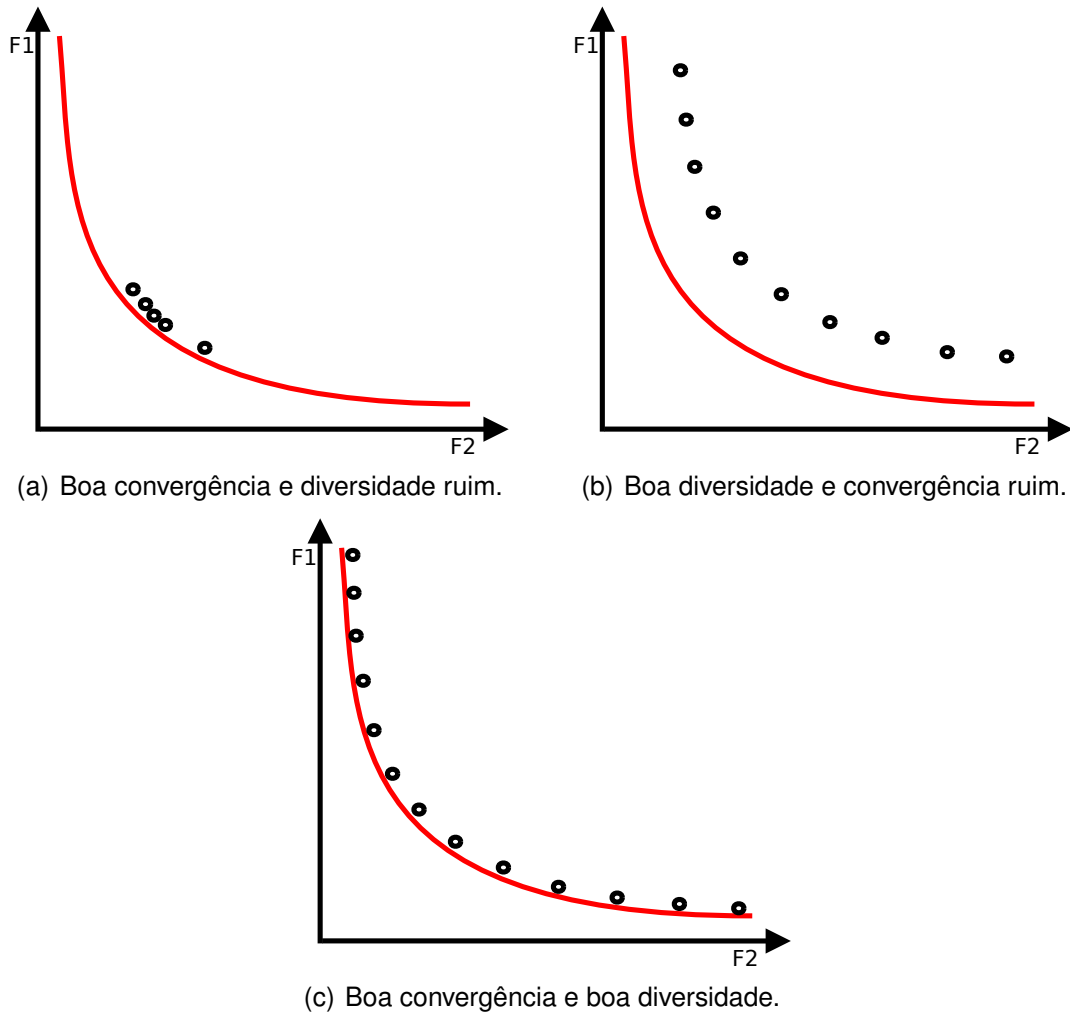


Figura 2.5: Exemplos com três conjuntos de soluções que ilustram relações entre convergência e diversidade.

Os algoritmos evolutivos multiobjetivo têm progredido constantemente. Pesquisadores de diversas áreas desenvolvem novas técnicas que são incorporadas aos MOEAs com o intuito de melhorar os resultados em termos de convergência e diversidade em relação à fronteira de Pareto. A intenção é obter fronteiras de Pareto que se aproximem cada vez mais às fronteiras de Pareto reais. A partir desse intuito, surge a necessidade avaliar o quão bons são os resultados dos conjuntos de soluções que são obtidos. Esta avaliação pode ser realizada através de métricas de desempenho. A seguir, são apresentadas as principais métricas utilizadas para avaliar o desempenho das soluções obtidas pelos MOEAs.

## 2.3 Métricas de desempenho para avaliar soluções de problemas com mais de um objetivo

Considerando que problemas com mais de um objetivo possuem diferentes conjuntos de soluções que não são dominadas, obtidos por diferentes algoritmos ou parâmetros, surge a questão [Yan et al., 2007]: quais conjuntos de soluções apresentam as melhores soluções ou qual conjunto de soluções apresenta resultados mais próximos à fronteira de Pareto real? A questão não é trivial e na literatura encontramos três aspectos que devem ser considerados para respondê-la:

- a distância entre o conjunto de soluções não dominadas e a fronteira de Pareto real deve ser minimizada;
- é desejável que o conjunto de soluções esteja uniformemente distribuído ao longo da fronteira;
- a extensão do conjunto obtido de soluções não dominadas deve ser maximizada.

O primeiro aspecto trata da qualidade das soluções e está associado à convergência da mesma, enquanto que os dois últimos consideram a distribuição do conjunto, uma propriedade chamada de diversidade das soluções. Na prática, essas duas propriedades são divergentes entre si, pois à medida que os resultados vão convergindo, soluções vão sendo eliminadas do conjunto final, empobrecendo a diversidade do mesmo. As métricas utilizadas para avaliar as soluções obtidas pelos algoritmos evolutivos atêm-se a três diretrizes fundamentais [Deb, 2011, Branke et al., 2008]:

1. avaliação da convergência para a fronteira Pareto ótima ou real, utilizando alguma taxa de erro ou distância de um conjunto de soluções referencial;
2. avaliação da cobertura das soluções ao longo da fronteira Pareto ótima, valendo-se de alguma medida de diversidade ao longo da curva da fronteira;
3. avaliação baseada na combinação entre convergência e diversidade das soluções.

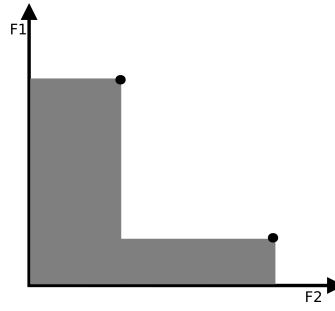


Figura 2.6: Exemplo da métrica *hypervolume* para um problema de maximização com dois objetivos [Zitzler, 1999].

A seguir, são apresentadas as principais métricas utilizadas na otimização multiobjetivo.

**a. Métrica *Hypervolume*:** apresentada por Zitzler [Zitzler, 1999], essa métrica considera o tamanho da área coberta pelas soluções no espaço de objetivos. Para calcular o *hypervolume* de um conjunto de soluções é necessário definir um ponto de referência. Para problemas de maximização, é comum definir a origem como referencial, enquanto que em problemas de minimização tipicamente utiliza-se o ponto nadir ( $z^{nadir}$ ). A Figura 2.6 ilustra um problema de maximização com dois objetivos em que a área formada pelo retângulo, definida pela origem e  $(f_1(\vec{x}), f_2(\vec{x}))$  determina o *hypervolume*. A métrica de convergência pode ser dada pela soma de todas as áreas dos retângulos definidos pelo conjunto das soluções.

**b. Métrica *Spacing*:** sugerida por Schott [Schott, 1995], a métrica *spacing* mede a faixa de variação das soluções vizinhas na fronteira de Pareto conhecida. A métrica é dada pela Equação 2.5.

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (2.5)$$

onde  $d_i = \min_j (|f_1(\vec{x}_i) - f_1(\vec{x}_j)| + |f_2(\vec{x}_i) - f_2(\vec{x}_j)|)$ ,  $\bar{d}$  é a média de todos os  $d_i$  e  $n$  é o número de soluções. Ao contrário da métrica *hypervolume*, a métrica *spacing* concentra-se na distância relativa entre as soluções e não da distância absoluta a partir da origem.



**c. Métrica *Generational Distance*:** Veldhuizen e Lamont [Van Veldhuizen e Lamont, 2002] propuseram uma medida que avalia a convergência do conjunto de soluções e determina quão longe a fronteira de Pareto conhecida está da fronteira de Pareto real. Este valor é determinado pela equação:

$$G = \frac{(\sum_{i=1}^n d_i^p)^{\frac{1}{p}}}{n} \quad (2.6)$$

onde  $n$  é o número de vetores na fronteira de Pareto conhecida,  $p = 2$  e  $d_i$  é a distância Euclidiana (no espaço de objetivo) entre cada vetor e o membro mais próximo da fronteira de Pareto real. Um resultado igual a 0 indica que a fronteira obtida e a fronteira real são iguais; qualquer outro valor indica que a fronteira conhecida desvia da fronteira real. O exemplo da Figura 2.7 possui  $d_1 = \sqrt{(2.5 - 2)^2 + (9 - 8)^2}$ ,  $d_2 = \sqrt{(3 - 3)^2 + (6 - 6)^2}$  e  $d_3 = \sqrt{(5 - 4)^2 + (4 - 4)^2}$ , sendo  $G = \sqrt{\frac{1.118^2 + 0^2 + 1^2}{3}} = 0.5$ .

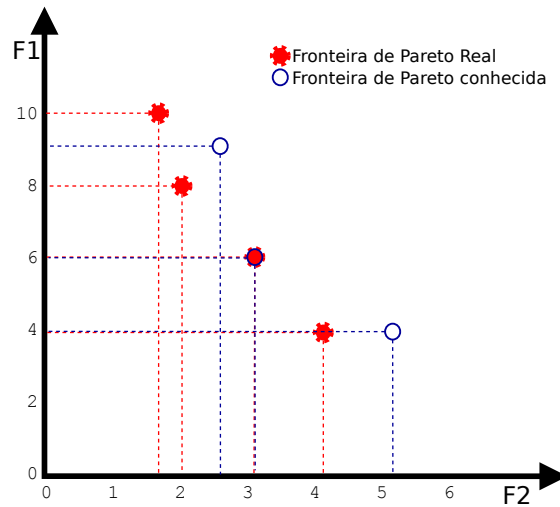


Figura 2.7: Exemplo de fronteiras de Pareto real e obtida para a métrica GD para um problema com dois objetivos de minimização [Van Veldhuizen e Lamont, 2002].

**b. Métrica *Inverse Generational Distance*:** apresentada por Coello e Cortés [Coello e Cortés, 2005], a métrica IGD mede a distância mínima entre cada ponto da fronteira de Pareto real em relação ao conjunto de soluções obtido.

O IGD permite observar se a fronteira de Pareto obtida converge para a fronteira de Pareto real e se este conjunto é diversificado. Quanto mais próximo a 0 seja

o valor do IGD, melhor o indicador.

**e. Métricas GDp e IGDp:** em 2012, Schütze *et al.* [Schütze et al., 2012] propuseram os indicadores GDp e IGDp. Tais indicadores são variações dos tradicionais indicadores *GD* - *Generational Distance* [Van Veldhuizen e Lamont, 2002] e *IGD* - *Inverted Generational Distance* [Coello e Cortés, 2005], respectivamente. Os autores argumentam que as métricas originais são fortemente influenciadas pelo número de soluções da fronteira analisada. Os indicadores acusam melhor desempenho em fronteiras que possuem mais pontos, mesmo que estes não aproximem-se da fronteira de Pareto real. Em outras palavras, o indicador é influenciado pelo número de pontos e não pela qualidade das soluções. Assim, os indicadores GDp e IGDp foram propostos para prover resultados mais justos quando compara conjuntos de soluções com diferentes magnitudes. São métricas especialmente adequadas para avaliar o desempenho de MOEAs que utilizam diferentes métodos para arquivamento de soluções.

Assim como o GD, o GDp avalia a convergência do conjunto de soluções e determina quão longe a fronteira de Pareto conhecida está da fronteira de Pareto real. Este valor é determinado pela Equação 2.6.

$$GDp = \left( \frac{1}{n} \sum_{i=1}^n dist(x_i, R)^p \right)^{\frac{1}{p}} \quad (2.7)$$

onde  $n$  é o número de vetores na fronteira de Pareto conhecida e  $dist(x_i, R)^p$  é a distância Euclidiana entre cada vetor e o membro mais próximo da fronteira de Pareto real. Um resultado igual a 0 indica que os pontos da fronteira obtida e da fronteira real coincidem; qualquer outro valor indica que a fronteira conhecida desvia da fronteira real. A Figura 2.8 ilustra um exemplo de cálculo do GDp em um problema com dois objetivos de minimização. Para cada ponto da fronteira conhecida é calculada a menor distância em relação à fronteira de Pareto real.

O IGDp, assim como o IGD faz o cálculo oposto ao adotado pelo GD. Ou seja, ele

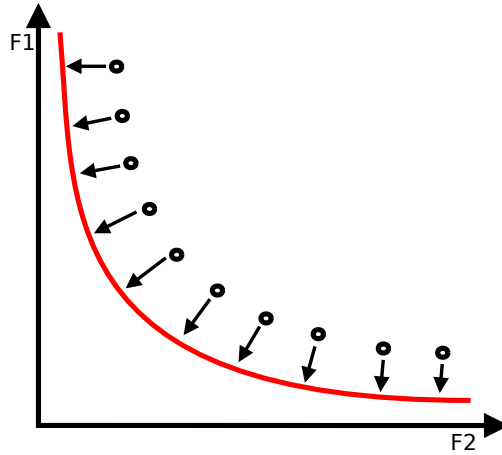


Figura 2.8: Exemplo de cálculo da métrica GDp: a linha representa a fronteira de Pareto real, enquanto que as setas correspondem à menor distância entre dois pontos entre as fronteiras conhecida e real.

mede a distância mínima entre cada ponto da fronteira de Pareto real em relação aos vetores da fronteira de Pareto conhecida. O IGDp possui a vantagem de permitir observar se a fronteira de Pareto conhecida converge para a fronteira de Pareto real e se este conjunto conhecido é diversificado. O valor do indicador IGDp é obtido pela Equação 2.8.

$$IGDp = \left( \frac{1}{n} \sum_{i=1}^n dist(r_i, X)^p \right)^{\frac{1}{p}} \quad (2.8)$$

A Figura 2.9 ilustra um exemplo de cálculo do IGDp em um problema com dois objetivos de minimização. Para cada ponto da fronteira de Pareto real é calculada a menor distância em relação à fronteira conhecida.

Não existe uma única métrica que seja adequada a todas as classes de problemas com mais de um objetivo. Todas possuem pontos negativos e positivos que já foram amplamente discutidos da literatura. Além das apresentadas, outras métricas são constantemente propostas. Assim, para avaliar a qualidade das soluções, deve-se considerar a utilização de mais de uma métrica para validar os resultados.

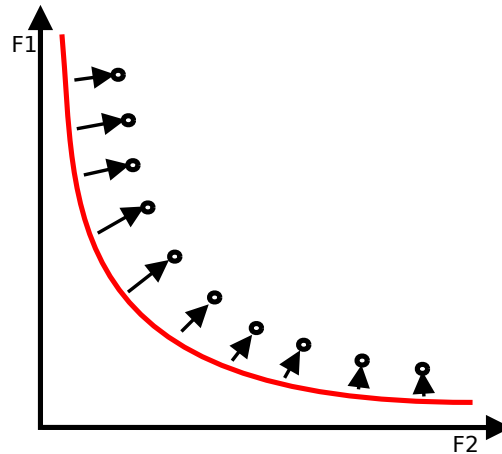


Figura 2.9: Exemplo de cálculo da métrica IGDp: a linha representa a fronteira de Pareto real, enquanto que as setas correspondem à menor distância entre dois pontos das fronteiras real e conhecida.

## 2.4 Otimização com vários objetivos

A otimização de problemas com mais de um objetivo é uma área de intensa pesquisa no campo da computação evolutiva [Coello et al., 2007, de Carvalho e Pozo, 2011a, Deb et al., 2002a, Ishibuchi et al., 2008, Mostaghim e Teich, 2003, Zhang e Li, 2007]. Vários algoritmos propostos para solucionar problemas com dois objetivos mostram-se adequados e eficientes. Entretanto, quando a otimização envolve mais que quatro objetivos, tais algoritmos têm seu desempenho degradado e por vezes mostram-se inadequados. Problemas multiobjetivo com quatro ou mais objetivos são frequentemente referidos como problemas de vários objetivos (MaOP, do inglês *Many-Objective Problem*) [Ishibuchi et al., 2009]. Nesse tipo de problema, surgem algumas dificuldades adicionais [Ishibuchi et al., 2008]:

1. A habilidade de busca é severamente deteriorada pelo aumento no número de objetivos. Isso ocorre, porque quase todas as soluções da população não são dominadas entre si. Como resultado, não existe pressão em direção à fronteira de Pareto e a busca é praticamente orientada de forma aleatória.

De modo a ilustrar a deterioração na habilidade de busca, a Figura 2.10 mostra o comportamento de um algoritmo evolutivo para resolver um problema à medida que o número de objetivos aumenta. Foi utilizado o algoritmo *Nondominated*

*Sorting Genetic Algorithm II* (NSGA-II) [Deb et al., 2002a] na resolução do problema DTLZ1 da família de *benchmarks* de problemas multiobjetivo DTLZ [Deb et al., 2002b], com 2, 5, 10, 20 e 30 objetivos. O algoritmo foi executado com parâmetros padrão [Durillo e Nebro, 2011] e para cada configuração de objetivos o algoritmo executou 25.000 gerações. Ao término da execução, o indicador *Generational Distance* (GD) [Van Veldhuizen e Lamont, 2002] foi utilizado para avaliar cada fronteira obtida. A intenção foi avaliar a convergência do conjunto de soluções e determinar quão distante a fronteira de Pareto obtida está da fronteira de Pareto real. Quanto menor a distância entre as duas fronteiras, melhor, pois indica que a fronteira de Pareto obtida está mais próxima da fronteira de Pareto real do referido problema. Através do gráfico da Figura 2.10 é possível observar o comportamento do algoritmo em relação à convergência entre as duas fronteiras. Nota-se que, com poucos objetivos o algoritmo possui capacidade de convergir em direção à fronteira de Pareto real. Porém, à medida que o número de objetivos aumenta, a convergência piora significativamente, evidenciando a deficiência do algoritmo evolutivo em lidar com muitos objetivos.

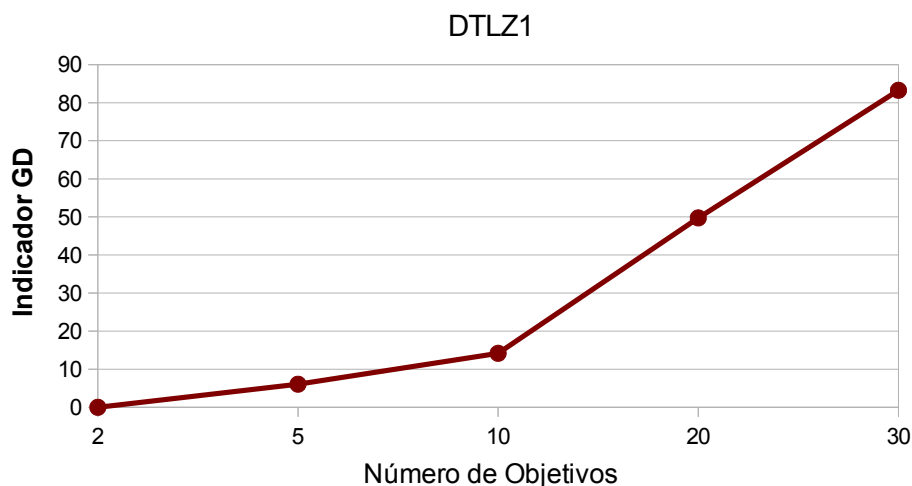


Figura 2.10: Deterioração da habilidade de busca do algoritmo NSGA-II na resolução do problema DTLZ1 em relação ao aumento no número de objetivos.

2. Com o aumento exponencial [Jaimes e Coello, 2009] no número de soluções não dominadas pela característica da fronteira de Pareto ser multidimensional, torna-



Figura 2.11: Visualização da fronteira de Pareto, segundo Obayashi e Sasaki [Obayashi e Sasaki, 2003].

se difícil definir uma estrutura de dados apta a trabalhar com um conjunto elevado de informações. Esta dificuldade foi estudada por Schutze *et al.* [Schutze et al., 2011], que investigaram a representação do conjunto de soluções à medida que o número de objetivos aumenta. Segundo os autores, considerando que  $k$  represente o número de objetivos de um problema multiobjetivo, tipicamente, teremos um conjunto de soluções  $(k-1)$ -dimensional. Assim,  $k$  exerce influência direta na dimensão do conjunto de Pareto: para  $k = 2$ , segundo os autores, tipicamente 100 pontos são “suficientes” para representar um conjunto de soluções. Para se obter a mesma qualidade de aproximação para  $k = 20$ , seriam necessários impraticáveis  $100^{19}$  elementos. Mesmo que fosse utilizada a quantidade mínima de 2 pontos ( $2^1 = 2$ ) para representar um conjunto de Pareto com  $k = 2$ , para 20 objetivos ( $k = 20$ ), seriam necessários  $2^{19} = 524.288$  elementos para obter a mesma (baixa) qualidade de aproximação.

3. Dificuldade na interpretação dos resultados, pois a visualização do conjunto de soluções gerado com muitos objetivos torna-se complexa. No contexto multiobjetivo, problemas com dois ou três objetivos permitem traçar gráficos com visualização intuitiva das soluções no espaço de busca. Porém, conforme ilustrado pela Figura 2.11 gráficos com mais de três dimensões não são desenhados de maneira usual. Assim, não é trivial para o tomador de decisão compreender quatro ou mais dimensões visualmente.

Para contornar essas dificuldades, muitos esforços têm sido realizados para permitir a escalabilidade dos problemas com vários objetivos.

Segundo Ishibuchi *et al.* [Ishibuchi et al., 2008], a primeira dificuldade pode ser contornada criando mecanismos para que as soluções sejam pressionadas em direção à fronteira de Pareto. Algumas estratégias adotadas consideram:

- modificar a relação de dominância de Pareto, de modo a diminuir ou aumentar a quantidade de soluções não dominadas [Sato et al., 2007, de Carvalho e Pozo, 2010, de Carvalho e Pozo, 2011a];
- ranquear as soluções criando uma classificação para as mesmas [Srinivas e Deb, 1994]. Baseado em critérios de ranqueamento, algumas soluções são preteridas em relação a outras;
- utilizar um repositório com as soluções não dominadas, estipulando um limite na quantidade de soluções não dominadas. Quando esse limite é atingido, métodos de arquivamento são empregados [de Britto et al., 2013];
- utilizar métodos alternativos à teoria de Pareto para avaliar a aptidão (*fitness*) das soluções. Esses métodos podem avaliar a qualidade da solução utilizando algum indicador de desempenho como GD, IGD ou *hypervolume* [Wagner et al., 2007].
- empregar métodos de decomposição dos problemas com vários objetivos em problemas mono-objetivo. Em seguida, MOEAs são utilizados para otimizar cada subproblema. O *Multiobjective Evolutionary Algorithm Based on Decomposition* (MOEA/D), desenvolvido por Zhang e Li [Zhang e Li, 2007] é um algoritmo baseado em decomposição de problemas bastante referenciado na literatura. Por este motivo, ele será tratado em detalhes na subseção 2.5.2.

Quanto à segunda dificuldade, segundo [Ishibuchi et al., 2008], uma estratégia que é frequentemente adotada diz respeito à incorporação de informações de preferência aos MOEAs. A informação de preferência é utilizada para definir pequenas regiões na fronteira de Pareto onde os algoritmos concentram-se para encontrar soluções não dominadas. Thiele *et al.* [Thiele et al., 2009] desenvolveram um algoritmo evolutivo

que incorpora informação de preferência fornecida por um tomador de decisão (TD). Ao associar valores às funções objetivo, o TD define pontos de referência que determinam partes da fronteira Pareto que serão priorizadas. Como consequência, pode-se manter uma população pequena. Uma inconveniência relacionada a esta proposta recai na necessidade do TD fornecer informações de preferência a cada iteração. Um algoritmo que restringe o número de interações requeridas junto ao TD foi proposto por [Sinha et al., 2012]. Os autores propuseram uma técnica que previamente determina alguns pontos como preferenciais e esporadicamente solicita a intervenção do TD. Os resultados obtidos foram confrontados com um algoritmo semelhante, mas que não limita a quantidade de chamadas ao TD. Os autores concluem que é possível obter um equilíbrio entre a quantidade de intervenções do TD e a precisão obtida nos resultados.

Finalmente, em relação à terceira dificuldade, múltiplos esforços têm sido empregados para representar visualmente os conjuntos de soluções obtidos para problemas com mais de três objetivos. Entre tais esforços, podemos citar: gráficos coordenados paralelos (*parallel coordinate plots*) [Fonseca e Fleming, 1993], gráficos coordenados em pares (*pairwise coordinate plots*) [Deb, 2001], mapa de calor (*heatmap*) [Pryke et al., 2006] e os mapas auto-organizáveis (SOM - *Self-Organising Maps*) [Kohonen, 1995]. É possível encontrar variações e combinações destas técnicas. Estudos aprofundados estão disponíveis em [Walker et al., 2013].

## 2.5 Algoritmos evolutivos para resolver MOPs

Como exposto na Seção 2.4, à medida que o número de objetivos aumenta, a habilidade de busca dos MOEAs se enfraquece. Esta propriedade é decorrente do fato de que a maioria das soluções da população não são dominadas entre si. Consequentemente, a pressão em direção à fronteira de Pareto fica debilitada. Vários algoritmos foram propostos para contornar essa dificuldade. Entre eles, dois merecem destaque por serem considerados referência na área e por exercer influência no desenvolvi-



mento dessa tese.

O primeiro algoritmo apresentado nesta seção é o NSGA-II (do inglês, *Nondominated Sorting Genetic Algorithm II*), desenvolvido por Deb *et al.* [Deb et al., 2002a] que insere um mecanismo de elitismo na busca. O segundo é o MOEA/D, desenvolvido por Zhang e Li [Zhang e Li, 2007] que decompõe problemas com vários objetivos em problemas mono-objetivo.

### 2.5.1 NSGA-II - *Nondominated Sorting Genetic Algorithm II*

O NSGA-II é um algoritmo genético estendido para encontrar múltiplas soluções de Pareto em um problema de otimização multiobjetivo [Deb et al., 2002a]. Este algoritmo possui as seguintes características: i) utiliza um princípio elitista; ii) utiliza um mecanismo explícito para preservar a diversidade; e. iii) enfatiza as soluções não dominadas. O Algoritmo 1 mostra o pseudo-código do NSGA-II. Inicialmente, o algoritmo gera aleatoriamente uma população  $P_0$ , com  $|P_0| = P$ . Através de operadores genéticos tradicionais (torneio, seleção, cruzamento ( $p_c$ ), e mutação ( $p_m$ )), os primeiros descendentes  $Q_0$  ( $|Q_0| = P$ ) são gerados. As populações  $P_0$  e  $Q_0$  são unidas e formam  $R_0$  ( $|R_0| = 2P$ ).

No processo iterativo  $t = 1, 2, 3, \dots, max\_it$ , o NSGA-II manipula a população  $R_t$ .  $R_t$  é ordenada através de sua não dominância. Isto é, inicialmente, todas as soluções não dominadas são colocadas na fronteira  $F_0$ . Após, os pontos de  $F_0$  são descartados do conjunto global de soluções (apenas para o procedimento de ordenação), e o novo conjunto de valores não dominados formam a fronteira  $F_1$ , e assim por diante.

O procedimento **calcule\_CrowdingDistance** é utilizado para formar a população  $P_{t+1}$ , baseado na densidade de soluções em volta de uma solução particular. Para o ponto de  $F_j$  (soluções da fronteira  $j$ ), o NSGA-II obtém sua densidade calculando a distância média entre dois pontos mais próximos ao longo de cada um dos objetivos do problema. O valor da medida CD é obtido somando os valores das distâncias obtidas. Esse procedimento retorna para cada solução a sua distância de agrupamento. Quanto maior o valor do CD de uma solução, menos densa será a área em que esta

---

**Algoritmo 1:** Pseudo-código do NSGA-II [Deb et al., 2002a].
 

---

**Entrada:**

Problema de Otimização;

 $max\_it \leftarrow$  número máximo de iterações.**Saída:** Conjunto de soluções não dominadas.

```

1:  $P_0 = Q_0 = \emptyset$ ;
2: Inicialize população  $P_0$ ;
3:  $contador = 0$  ;
4: Inicialize o número de gerações;
5: Seleção por torneio;
6: Cruzamento;
7: Mutação;
8: Gere População  $Q_0$ ;
9: enquanto  $contador < max\_it$  faça
10:    $R_t = P_t \cup Q_t$ ;
11:   Ordene  $R_t$  por sua não dominância;
12:    $P_{t+1} = \emptyset$ ;
13:   enquanto  $|P_{t+1}| \leq P$  faça
14:     Calcule  $CrowdingDistanceF_j$ ;
15:      $P_{t+1} = P_{t+1} \cup F_j$ ;
16:   fim-enquanto
17:   Ordene  $P_{t+1}$  usando o operador  $\geq_t$ ;
18:    $P_{t+1} = P_{t+1}[0 : (P - 1)]$ ; Escolha os primeiros  $P$ ;
19:   Cruzamento;
20:   Mutação;
21:   Gere a população  $Q_{t+1}$ ;
22:    $contador = contador + 1$ ;
23: fim-enquanto

```

---

solução está localizada.

Durante a seleção, o NSGA-II utiliza um operador de comparação de densidade que leva em conta tanto o ranqueamento individual (em relação aos critérios de não dominância) quanto sua distância de agrupamento. As soluções não dominadas são preferenciais em relação às soluções dominadas. Para soluções com a mesma ordem no ranqueamento de soluções não dominadas, a solução localizada em uma região menos populosa é mantida. Assim, somente as melhores  $P$  soluções de  $P_{t+1}$  permanecem na população. Finalmente, os operadores genéticos são aplicados em  $P_{t+1}$  para formar os novos descendentes  $Q_{t+1}$ . A Figura 2.12 ilustra os procedimentos adotados pelo algoritmo NSGA-II.

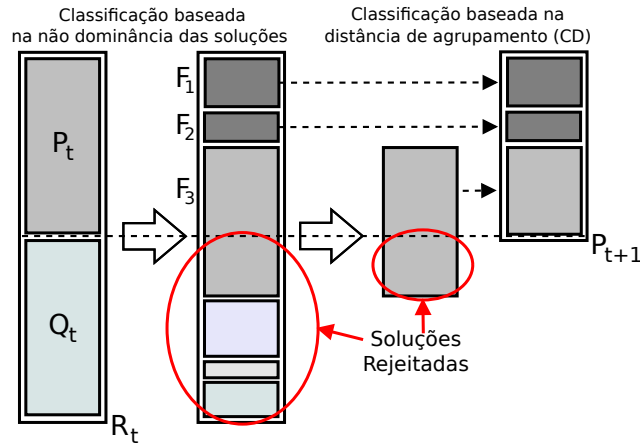


Figura 2.12: Procedimentos do NSGA-II [Deb et al., 2002a].

### 2.5.2 MOEA/D - *Multiobjective Evolutionary Algorithm Based on Decomposition*

Zhang e Li [Zhang e Li, 2007] propuseram um algoritmo evolutivo baseado em decomposição para solucionar problemas multiobjetivos chamado de MOEA/D (do inglês, *Multiobjective Evolutionary Algorithm Based on Decomposition*). A contribuição deste algoritmo na área foi de tamanha relevância, que Coello e Martínez afirmam em [Zapotecas Martínez e Coello Coello, 2011] que ele deu origem a uma nova geração de MOEAs. Deste então, algoritmos inspirados no MOEA/D têm apresentado resultados promissores [Peng e Zhang, 2008, Moubayed et al., 2010, Zapotecas Martínez e Coello Coello, 2011, Venske et al., 2012, Wang et al., 2013, Al Moubayed et al., 2012, Ke et al., 2013].

O MOEA/D decompõe um MOP em  $N$  problemas escalares de otimização mono-objetivo. Ele resolve esses problemas simultaneamente evoluindo uma população de soluções. Para cada solução, o processo considera alguns indivíduos vizinhos, estabelecendo uma forma de co-evolução. A cada geração, a população é composta da melhor solução encontrada no momento. Assim, um conjunto de soluções aproximadas ao ótimo de Pareto é obtido otimizando cada subproblema (função de agregação escalar) individualmente. De acordo com Messac *et al.* [Messac et al., 2003], é desejável que os métodos para a construção de funções de agregação possuam os seguintes atributos:

1. o método deve gerar uma frente de Pareto bem distribuída e não negligenciar nenhuma região;
2. o método deve ter a habilidade de gerar todas as soluções de Pareto disponíveis;
3. o método deve gerar somente soluções de Pareto não dominadas;
4. o método deve ser relativamente simples de ser aplicado.

Muitos métodos para a construção de tais funções podem ser utilizados, porém, os mais frequentes são os métodos de: Soma Ponderada, *Tchebycheff* e *Boundary Intersection* (BI) [Miettinen, 1999, Zhang e Li, 2007]. A seguir, essas estratégias são discutidas considerando o problema de minimização apresentado na Equação 2.2.

### **Método de soma ponderada (*Weighted Sum Approach*) [Miettinen, 1999]**

Este método considera uma combinação de objetivos com importâncias distintas. Seja  $\lambda = (\lambda_1, \dots, \lambda_m)^T$  um vetor de pesos, i.e.,  $\lambda_i \geq 0$  para todo  $i = 1, \dots, m$  e  $\sum_{i=1}^m \lambda_i = 1$ . Então, a solução ótima para o seguinte problema de otimização escalar:

$$\begin{aligned} \text{minimizar } g^{ws}(\vec{x}|\vec{\lambda}) &= \sum_{i=1}^m \lambda_i f_i(\vec{x}), \\ \text{sujeito a } \vec{x} &\in \Omega \end{aligned} \tag{2.9}$$

é um ponto ótimo de Pareto da função de minimização 2.2, onde  $g^{ws}(x|\lambda)$  é usado para enfatizar que  $\vec{\lambda}$  é um vetor de coeficientes nesta função de objetivos e  $\vec{x}$  são as variáveis a serem otimizadas. Para gerar um conjunto diferente de vetores ótimos de Pareto, diferentes vetores de peso  $\vec{\lambda}$  podem ser empregados. Entretanto, nem todos os vetores ótimos de Pareto podem ser obtidos por este método caso a curva da função seja não côncava. Estratégias para contornar esta limitação estão disponíveis em [Miettinen, 1999].

## Método de *Tchebycheff* [Miettinen, 1999]

Neste método, o problema de otimização escalar mono-objetivo está na forma

$$\begin{aligned} \text{minimizar } g^{Tchebycheff}(\vec{x}|\vec{\lambda}, \vec{z}^*) &= \max_{1 \leq i \leq m} \{\lambda_i |f_i(\vec{x}) - z_i^*|\}, \\ \text{sujeito a } x &\in \Omega \end{aligned} \quad (2.10)$$

onde  $\vec{z}^* = (z_1^*, \dots, z_m^*)^T$  é o ponto de referência, i.e.,  $z_i^* = \min\{f_i(\vec{x}) | x \in \Omega\}$  para cada  $i = 1, \dots, m$ . Para cada ponto ótimo de Pareto  $x^*$  existe um vetor de peso  $\lambda$ , tal que  $x^*$  é a solução ótima de 2.10 e cada solução ótima de 2.10 é um ponto ótimo de Pareto da função de minimização 2.2. Diferentes soluções ótimas de Pareto podem ser obtidas ao se modificar o vetor de pesos. Uma desvantagem dessa função de agregação é a dificuldade em cobrir curvas suaves de MOPs contínuos.

## Método de interseção de limites (*Boundary Intersection* - BI) [Das e Dennis, 1998]

Métodos BI foram projetados para MOPs contínuos. Normalmente, a fronteira de Pareto de um MOP contínuo de minimização é o limite da curva à esquerda e abaixo da região factível do conjunto de objetivos. Este método objetiva encontrar pontos de interseção na maior parte dessa curva. Nesse método, o problema de otimização escalar mono-objetivo está na forma

$$\begin{aligned} \text{minimizar } g^{bi}(\vec{x}|\vec{\lambda}, \vec{z}^*) &= d \\ \text{sujeito a } F(x) - z^* &= d\lambda, \\ \vec{x} &\in \Omega \end{aligned} \quad (2.11)$$

onde  $\vec{\lambda}$  e  $\vec{z}^*$  são, respectivamente, vetores de peso e o ponto de referência. O objetivo é conduzir  $F(\vec{x})$  em direção à curva da região factível da fronteira. Embora a restrição da Equação 2.11 possa ser uma dificuldade, este método apresenta duas vantagens significativas em relação ao método de *Tchebycheff*: 1) tendência das soluções resultantes serem mais bem distribuídas; e 2) possibilidade rara de gerar

---

**Algoritmo 2:** Pseudo-código do MOEA/D [Zhang e Li, 2007].

---

**Entradas:**

- Problema de Otimização;
- um critério de parada;
- $N$ : número de subproblemas;
- uma distribuição uniforme de  $N$  vetores de pesos:  $\lambda^1, \dots, \lambda^N$ ;
- $T$ : número de vetores de peso vizinhos de cada vetor de peso.

**Saída:** Soluções não dominadas do arquivo externo  $A$ .

**1: ETAPA 1: Inicialização:**

2:  $A \leftarrow \emptyset$ ;

3: Calcule a distância Euclidiana entre todos os pares de vetores de pesos e encontre os  $T$  vetores de peso mais próximos de cada vetor de peso. Para cada  $i = 1, \dots, N$ ,  $B(i) = \{i_1, \dots, i_T\}$ , onde  $\lambda^{i_1}, \dots, \lambda^{i_T}$  são os  $T$  vetores de pesos mais próximos a  $\lambda^i$ ;

4: Gere aleatoriamente a população inicial  $x^1, \dots, x^N$  e defina  $FV^i = F(x^i)$ ;

5: Inicie  $z : z_j = \min_{\text{objetivo } j}(FV^i), i = 1, \dots, N, j = 1, \dots, T$ .

**6: ETAPA 2: Atualização:**

7: **para**  $i = 1, \dots, N$  **faça**

8: Selecione aleatoriamente dois índices  $k, l$  de  $B(i)$  e gere uma nova solução  $y$  a partir de  $x^k$  e  $x^l$  usando operadores genéticos;

9: Aplique alguma heurística de melhoramento e/ou reparação em  $y$  e gere  $y'$ ;

10: Atualize  $z$ : para cada  $j = 1, \dots, m$ , se  $z_j < f_j(y')$ , então  $z_j = f_j(y')$ ;

11: Atualize as soluções vizinhas: para cada índice  $j \in B(i)$ , se  $g^{Tchebycheff}(y'|\lambda^j, z) \leq g^{Tchebycheff}(x^j|\lambda^j, z)$ , então  $x^j = y'$  e  $FV^j = F(y')$ ;

12: Atualize o arquivo externo  $A$ :

13: remova do repositório  $A$  todas as soluções dominadas por  $F(y')$ ;

14: adicione  $F(y')$  ao repositório  $A$  se nenhuma solução de  $A$  domina  $F(y')$ .

15: **fim-para**

**16: ETAPA 3: Critério de Parada:**

17: Se o critério de parada for satisfeito, então pare e mostre o arquivo externo  $A$ .

Caso contrário volte à **ETAPA 2**

---

a incorreta situação  $g^{Tchebycheff}(\vec{x}|\vec{\lambda}, \vec{z}^*) = g^{Tchebycheff}(\vec{y}|\vec{\lambda}, \vec{z}^*)$  mesmo que  $x$  domine  $y$ .

Qualquer dos métodos apresentados pode ser usado para decompor a aproximação da fronteira de Pareto em um número de problemas escalares de otimização. O MOEA/D utiliza o método de *Tchebycheff* para decompor um MOP e seu pseudo-código é apresentado no Algoritmo 2.

Na inicialização, baseada em suas distâncias Euclidianas,  $B(i)$  contém os  $T$  índices dos vetores mais próximos de  $\lambda^i$ . Se  $j \in B(i)$ , o  $j_{esimo}$  subproblema pode ser considerado como um vizinho do  $i_{esimo}$  subproblema. No  $i_{esimo}$  passo do laço da ETAPA 2, os  $T$  subproblemas vizinhos do  $i_{esimo}$  subproblema são considerados. Sendo  $x^k$  e  $x^l$  as atu-

ais melhores soluções vizinhas ao  $i_{esimo}$  subproblema, então o descendente de  $y$  tende a ser uma boa solução para o  $i_{esimo}$  subproblema. Os operadores genéticos utilizados são: i) operador de cruzamento de um ponto; e ii) operador de mutação padrão. O cruzamento de um ponto é, inicialmente, aplicado em duas soluções e gera uma solução filho. Então, um operador de mutação padrão é aplicado em cada posição da solução filho com uma probabilidade de 1%, gerando a solução  $y$ . No próximo passo (linha 9 do pseudo-código), heurísticas podem ser utilizadas para melhorar ou reparar  $y$ , gerando  $y'$ . Isso garante que  $y'$  seja uma solução factível. Na linha 10 o vetor de referência é atualizado e, no passo a seguir, todos os vizinhos do  $i_{esimo}$  subproblema são considerados e  $x^j$  é substituído por  $y'$  se  $y'$  for melhor que  $x^j$  em relação ao  $j_{esimo}$  subproblema.  $FV^j$  é necessário para a computação do valor de  $g^{Tchebycheff}(x^j|\lambda^j, z)$  da linha 11 e  $FV^i$  representará o melhor vetor objetivo associado com o  $i_{esimo}$  vetor de pesos.

O potencial observado no MOEA/D para resolver MOPs motivou o desenvolvimento de adaptações de seu *framework* para operadores não genéticos [Venske et al., 2012, Wang et al., 2013], inclusive no contexto da otimização por enxame de partículas [Peng e Zhang, 2008, Moubayed et al., 2010, Zapotecas Martínez e Coello Coello, 2011, Al Moubayed et al., 2012]. Considerando o escopo desta tese, o Capítulo 3 abordará essa metaheurística para prover um embasamento necessário ao entendimento do tema. Algoritmos baseados em decomposição de problemas utilizando o algoritmo PSO serão discutidos na Subseção 4.4.1 do Capítulo 4.

## CAPÍTULO 3

### A OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Neste capítulo são abordados os principais conceitos relacionados à otimização por enxame de partículas. Inicialmente, seu algoritmo é descrito incluindo as principais modificações para melhorar seu desempenho. A seguir, as principais topologias de vizinhança são descritas e o conceito de múltiplos e cooperativos enxames é discutido. Os principais algoritmos disponíveis na literatura são analisados e uma taxonomia pra identificar e classificar as estratégias de otimização cooperativa é apresentada. Finalmente, são expostas as principais estratégias para a distribuição e comunicação dos enxames.

#### 3.1 Otimização por enxame de partículas

A Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) é uma meta-heurística desenvolvida por Kennedy e Eberhart [Kennedy e Eberhart, 1995] que surgiu a partir da observação do comportamento social da movimentação de agrupamento de animais, por exemplo revoadas de pássaros e cardumes de peixes. No algoritmo, cada indivíduo é chamado de partícula e se comporta como um pássaro do bando a procura de alimento ou fugindo de algum predador. Cada partícula possui uma posição e uma velocidade de deslocamento usada para percorrer o espaço de soluções do problema. Uma posição está vinculada à solução do problema e corresponde a uma solução em potencial. Um conjunto de partículas é chamado de enxame, população ou *swarm* e realiza uma busca cooperativa em regiões consideradas promissoras no espaço de busca. Para atingir seu objetivo, a partícula utiliza o aprendizado adquirido por suas próprias experiências (informação ou componente individual) e também o aprendizado do bando (informação ou componente social). A melhor solução encontrada pela própria partícula até o momento é chamada de *pbest*



**Algoritmo 3:** Pseudo-código do PSO.

---

```

1: para cada partícula  $i$  faça
2:   inicializar  $\vec{x}_i, \vec{v}_i$ ;
3: fim-para
4: enquanto número máximo de iterações ou erro mínimo não atingido faça
5:   para cada partícula  $i$  faça
6:     calcule o valor atual de aptidão  $f(\vec{x}_i)$ ;
7:     se  $f(\vec{x}_i)$  é melhor que  $f(pbest_i)$  então
8:        $pbest_i \leftarrow \vec{x}_i$ ;
9:     fim-se
10:   fim-para;
11:    $gbest \leftarrow Melhor(pbest_i), i=1$  até  $N$ ;
12:   para cada partícula  $i$  faça
13:     calcule a velocidade  $\vec{v}_i$  de acordo com a equação 3.2;
14:     atualize a posição  $\vec{x}_i$  de acordo com a equação 3.1;
15:   fim-para
16: fim.

```

---

(valor individual) e a melhor solução encontrada pelo grupo é chamada  $gbest$  (valor global). O pseudocódigo do PSO é apresentado no Algoritmo 3.

Conforme apresentado no Algoritmo 3, o primeiro passo consiste em iniciar, aleatoriamente os valores de cada partícula  $\vec{x}_i$  e sua velocidade  $\vec{v}_i$  dentro dos limites definidos para cada função. A seguir, é executado um laço em que, para cada partícula, calcula-se sua aptidão (*fitness*). O valor da melhor solução é armazenado em seu respectivo  $pbest$  (no código,  $pbest_i$ ). A seguir, o melhor de todos os  $pbest$  é associado à variável  $gbest$  (melhor global). Após encontrar os dois melhores valores ( $pbest$  e  $gbest$ ), cada partícula deve atualizar a sua velocidade e a sua posição. Essas atualizações permitem que a partícula se “movimente” no espaço de busca em direção aos melhores resultados de um determinado problema. Cada partícula  $i$  possui uma posição  $\vec{x}(t) \in \mathbb{R}^n$  em um determinado tempo  $t$ . Esse tempo é expresso pela iteração no processo de execução. A nova posição de uma partícula  $i$ , em um tempo  $t+1$ , é calculada incluindo-se a velocidade ( $\vec{v}(t) \in \mathbb{R}^n$ ) à posição  $\vec{x}(t)$ , de modo que:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \quad (3.1)$$

Quanto à atualização da velocidade da partícula  $i$ , no tempo  $t+1$ , ela é determinada

por:

$$\vec{v}_i(t+1) = \vec{v}_i(t) + C1 \cdot rand() \cdot (p\vec{best}_i(t) - \vec{x}_i(t)) + C2 \cdot rand() \cdot (g\vec{best}(t) - \vec{x}_i(t)) \quad (3.2)$$

onde,  $C1$  e  $C2$  são constantes que definem o fator de aprendizagem e  $rand()$  é um número aleatório entre 0 e 1.  $\vec{v}_i$  representa a velocidade da partícula;  $p\vec{best}_i$ , o melhor valor encontrado pela própria partícula;  $\vec{x}_i$  corresponde à partícula sendo manipulada e  $g\vec{best}$  é a melhor solução encontrada pelo bando. Para evitar que a velocidade ultrapasse o espaço de busca, ela pode ser limitada a um valor máximo ( $vmax$ ). Se a velocidade exceder o valor, ela será limitada a  $vmax$ . A cada passo, o resultado tende a convergir para o resultado esperado. As melhores partículas da população são definidas através de uma função de *fitness*, que é a função objetivo do problema e é condicionada à posição da partícula. Considerando que  $f$  representa o *fitness* de uma partícula  $i$ ,  $f : S \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ , para problemas de minimização, a partícula  $i$  é melhor que a partícula  $j$ , se  $f(\vec{x}_i(t)) < f(\vec{x}_j(t))$ . O inverso é verdadeiro para problemas de maximização. Ao final de um limite de iterações ou algum critério de parada, a melhor solução é apresentada como resultado.

### 3.2 Modificações na fórmula de atualização da velocidade da partícula do algoritmo PSO

A otimização por enxame de partículas é uma área de intensa pesquisa, cujo algoritmo recebeu diversas modificações para melhorar o seu desempenho. Tais modificações têm como objetivo melhorar a habilidade de pesquisa no espaço das variáveis do problema. A seguir, são apresentadas duas modificações na fórmula de atualização da velocidade da partícula que são bastante utilizadas na literatura [Eberhart e Shi, 2000].

### 3.2.1 Equação da velocidade com peso da inércia $\omega$

Conforme constatado por Shi e Eberhart [Shi e Eberhart, 1998], a equação de atualização da velocidade da partícula pode ser dividida em três partes: a primeira parte compreende a velocidade anterior da partícula e as segunda e terceira partes correspondem às contribuições para modificar a velocidade. Segundo os autores, cada parte da equação influencia na habilidade de busca local ou na habilidade de explorar novos espaços de busca. Como o resultado final da solução é altamente dependente da semente que gerou a população inicial, alguns problemas serão beneficiados pela busca local enquanto outros serão beneficiados pela busca global. Assim, deve haver um equilíbrio entre essas habilidades para a resolução de diferentes problemas, isto é, deve existir um balanceamento em tais habilidades. Posto isso, os autores introduzem um novo parâmetro à formula original, chamado de peso de inércia, expresso pela variável  $\omega$ . Esse parâmetro compreende um valor positivo que pode se modificar ao longo do tempo. A fórmula modificada para atualização da partícula, considerando o peso da inércia é mostrada na equação 3.3.

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + C1 \cdot rand() \cdot (\vec{pbest}_i(t) - \vec{x}_i(t)) + C2 \cdot rand() \cdot (\vec{gbest}_i(t) - \vec{x}_i(t)) \quad (3.3)$$

Desse modo, é incorporado um parâmetro que denota a tendência da partícula manter a velocidade ao ter seu valor atualizado. A variável de inércia é linearmente reduzida de um valor inicial até um valor final, comumente entre 0,9 e 0,4, respectivamente [Zhang, 2011]. Experimentos mostram que o uso desse parâmetro melhora o desempenho do algoritmo PSO e possibilita que os resultados converjam para o ótimo global com mais eficiência.

### 3.2.2 Equação da velocidade com fator de constrição

Outra modificação na equação original de atualização da velocidade da partícula foi introduzida por Clerc [Clerc, 1999]. Nesse trabalho, o autor propõe a aplicação de um

fator de constrição na equação. Esse fator pode ser necessário para garantir a convergência do algoritmo PSO e é representado por  $K$  que multiplica toda a velocidade (equação 3.2).  $K$  é definido em função dos coeficientes  $C1$  e  $C2$  da equação de atualização da velocidade da partícula. Se  $C1 + C2 \leq 4$ , então  $K = 1$ , caso contrário, o valor de  $K$  será determinado pela equação 3.4:

$$K = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4 \cdot \varphi}|} \quad (3.4)$$

Onde:

$$\varphi = C1 + C2. \quad (3.5)$$

A equação 3.6 apresenta a fórmula para a atualização da velocidade da partícula com o fator de constrição:

$$\vec{v}_i(t+1) = K \cdot \left( \vec{v}_i(t) + C1 \cdot rand() \cdot (\vec{pbest}_i(t) - \vec{x}_i(t)) + C2 \cdot rand() \cdot (\vec{gbest}_i(t) - \vec{x}_i(t)) \right) \quad (3.6)$$

Esse mecanismo assegura que a velocidade da partícula não cresça de maneira ilimitada, mantendo a velocidade a patamares que permitam que a partícula explore mais eficientemente o espaço de busca.

### 3.3 Topologias de vizinhança

Na otimização por enxame de partículas, um enxame é um conjunto de partículas que se move em um espaço de busca n-dimensional. Como observado pela equação 3.2 da seção anterior, seu deslocamento no espaço de busca é determinado pela melhor solução encontrada pela partícula e pela melhor posição alcançada por seus vizinhos. A definição da melhor solução da vizinhança depende da topologia de vizinhança que conecta os indivíduos do enxame. Essa topologia estabelece uma relação na maneira como as partículas pertencentes a um enxame trocarão entre si a informação ou componente social. Kennedy [Kennedy, 1999] considera quatro topologias princi-

pais, conforme a vizinhança das partículas: anel, estrela, grafo completo e arbitrária. Destaca-se que em [Kennedy, 1999] a topologia conecta indivíduos de um mesmo enxame e não múltiplos enxames. Outras topologias são possíveis, diversificando as disposições das arestas. As partículas podem ser conectadas umas às outras através de qualquer topologia de vizinhança que possa ser representada como um grafo. A seguir, são apresentadas as topologias de vizinhança mais utilizadas em PSO [Reyes-sierra e Coello, 2006]:

- **grafo desconectado:** nesta topologia, todas as partículas estão isoladas. Cada partícula está conectada a si própria e compara sua posição atual apenas à sua melhor posição conhecida até então.
- **grafo completo:** no grafo completo, ilustrado pela Figura 3.1, cada partícula está conectada a todas as outras partículas. Cada partícula utiliza a sua própria experiência através do componente individual ( $pbest$ ) e utiliza a posição da melhor partícula de todo o enxame ( $gbest$ ).

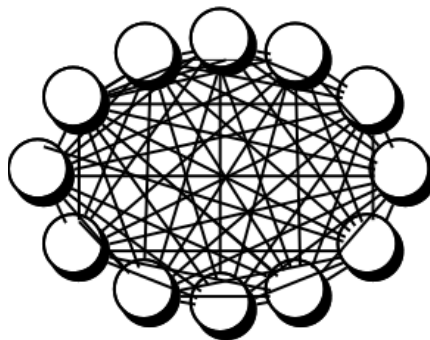


Figura 3.1: Topologia de vizinhança em forma de grafo completo.

- **melhor local:** nesta topologia, cada partícula é afetada pelas melhores informações dos  $k$  vizinhos imediatos. Cada partícula, além de utilizar sua própria experiência ( $pbest$ ), é influenciada pelas melhores posições dentro desta vizinhança (um melhor local ou  $lbest$ ).
- **anel:** também baseada em um melhor local, esta topologia possui dois vizinhos imediatos. Nesse caso,  $k = 2$ . A Figura 3.2 ilustra essa topologia.

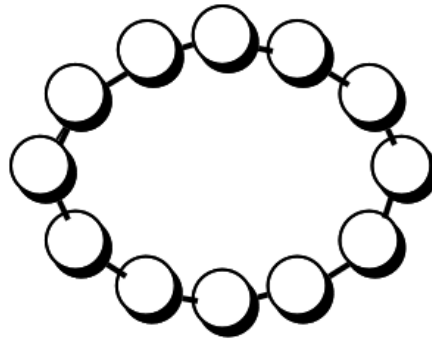


Figura 3.2: Topologia de vizinhança em forma de anel.

- **estrela:** nesta topologia, uma única partícula (chamada de partícula focal) é conectada a todas as outras partículas e cada partícula é conectada a apenas uma. Conforme ilustrado pela Figura 3.3, as partículas são isoladas umas das outras e toda a comunicação ocorre através da partícula focal. Essa partícula compara os resultados de todas as partículas do enxame e ajusta sua trajetória em direção à partícula com o melhor resultado. Eventualmente, este resultado pode ser comunicado ao restante do enxame.

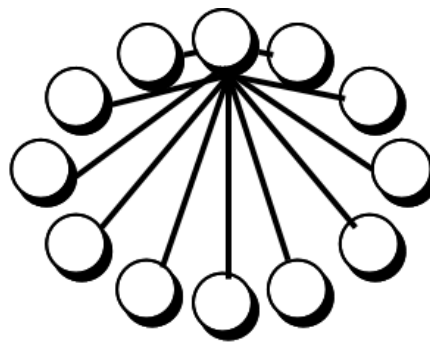


Figura 3.3: Topologia de vizinhança em forma de estrela.

- **árvore:** também chamada de topologia hierárquica, todas as partículas são arranjadas em uma árvore em que cada nodo da árvore contém apenas uma partícula, conforme ilustrado pela Figura 3.4. Uma partícula é influenciada por sua própria experiência através do componente individual (*pbest*) e pela melhor posição da partícula que está imediatamente acima na árvore (nodo pai). Caso uma partícula de um nodo filho encontre uma solução que seja melhor que a

solução da partícula do nodo pai, então ambas partículas são trocadas entre si.

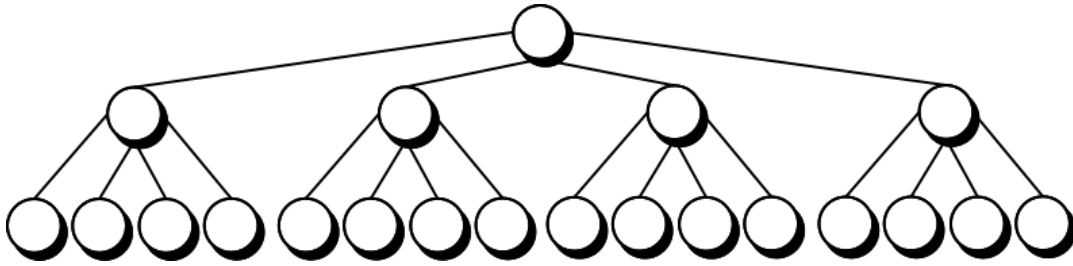


Figura 3.4: Topologia de vizinhança em forma de árvore.

- **arbitrária:** ilustrada pela Figura 3.5, esta topologia simplesmente associa conexões aleatórias entre pares de partículas. Comumente, para  $n$  partículas, são associadas  $n$  conexões [Kennedy, 1999].

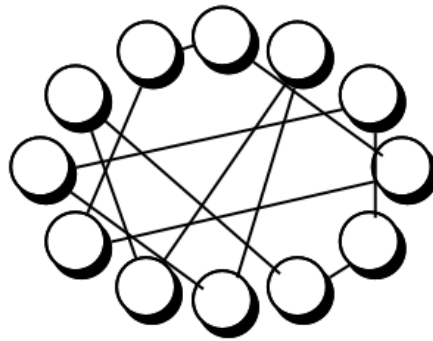


Figura 3.5: Topologia de vizinhança de forma arbitrária.

A relação entre as partículas de um enxame - determinada pela topologia estabelecida - é de fundamental importância para o desempenho do algoritmo. Topologias cujas partículas estão completamente conectadas permitem que cada partícula receba a informação da melhor solução de todo o enxame. Assim, ao se utilizar essa topologia, o enxame tende a convergir mais rapidamente do que se utilizasse uma topologia baseada em ótimo local (*lbest*), pois a melhor posição leva mais tempo para ser transferida. Todavia, em algumas situações, a topologia em questão é mais suscetível de sofrer convergência prematura, isso é, estagnar em ótimos locais [Campos Jr. et al., 2011]. O uso de múltiplos enxames na Otimização por Enxame de Partículas é uma alternativa para atacar esse problema e é tratado na próxima seção.

### 3.4 PSO com múltiplos enxames

Como dito anteriormente, na literatura é possível encontrar muitas variações do algoritmo PSO com o intuito de melhorar seu desempenho. Dentre essas variações, uma que tem se destacado consiste na divisão das partículas em subpopulações ou múltiplos enxames (*swarms*), caracterizando um modelo de otimização cooperativo. Em tais modelos, cada enxame pode executar o processo de otimização de maneira independente. Nesse caso, a execução em paralelo dos múltiplos enxames traz como benefício mais evidente a melhora no tempo necessário para realizar o processo de otimização. Outra vantagem, observada por [LÃ\_vbjerg et al., 2001], é que esse modelo apresenta-se como uma tentativa para evitar que o algoritmo convirja para sub-ótimos (ou falsos ótimos globais) precocemente. Muitos trabalhos têm explorado a cooperação de múltiplos enxames na otimização por enxame de partículas de diversas maneiras.

#### 3.4.1 *Multiple Swarm PSO - MSPSO*

O primeiro trabalho associando o modelo de otimização cooperativo com o algoritmo PSO foi proposto por Van der Bergh e Engelbrecht [Rakitienskaia e Engelbrecht, 2008] que desenvolveram os algoritmos  $CPSO - S_k$  e  $CPSO - H_k$  (CPSO, do inglês *Cooperative Particle Swarm Optimizer*). O algoritmo  $CPSO - S_k$  divide o vetor de solução em vetores menores ( $k$  enxames) e cada enxame é otimizado independentemente. A cada iteração, a solução é montada a partir das melhores partículas de cada enxame, reconstruindo, assim, o vetor de solução. A atualização de cada enxame envolve a melhor partícula do próprio enxame e a melhor partícula de todos os enxames. O  $CPSO - H_k$  é uma implementação híbrida que combina o primeiro algoritmo ao algoritmo PSO padrão. Os autores têm a intenção de combinar as melhores características dos dois algoritmos em uma solução. A ideia consiste em criar um ciclo onde algumas partículas de um algoritmo são substituídas pelas melhores partículas do outro. Entretanto, ambos algoritmos CPSO são dependentes do tipo de função que se es-



teja otimizando. Como observado por Li e Yao [Li e Yao, 2009, Li e Yao, 2011], visto que existe uma correlação entre os enxames, pois cada enxame contém parte do vetor solução, funções que não podem ser separadas têm o desempenho degradado nos dois algoritmos CPSO. Outra questão a ser observada é que para cada parte do problema dividido são avaliadas as melhores partículas locais e globais, aumentando significativamente o custo computacional.

Outro trabalho envolvendo múltiplos enxames foi apresentado por Liang e Suganthan [Liang e Suganthan, 2005] em que os autores apresentam o algoritmo DMS-PSO (do inglês, *Dynamic Multi-Swarm - PSO*) que tem como principal característica a adoção de uma estrutura de vizinhança dinâmica entre múltiplos enxames. Este dinamismo é introduzido através de um mecanismo de reagrupamento realizado em um intervalo pré-determinado. Este reagrupamento é realizado aleatoriamente e garante que as partículas pertençam a diferentes enxames. É constatado que poucas partículas por enxame podem alcançar melhores resultados para problemas simples, enquanto que, para problemas complexos uma pequena vizinhança entre os enxames é mais adequada. Nesta configuração, a velocidade de convergência diminui e a diversidade aumenta. São executadas avaliações em funções de *benchmarks* e os autores concluem que os melhores resultados são obtidos em uma configuração com 30 partículas, três partículas por enxame e reagrupamento realizado a cada 5 iterações. Entretanto, os experimentos são realizados na resolução de problemas com apenas dez dimensões, considerados pouco complexos.

Em [Zheng et al., 2007] é proposto o CCPSO (do inglês, *Cooperative Coevolutionary PSO*) que incorpora um operador de mutação e expõe um método alternativo para a escolha de indivíduos de cada subpopulação que comporá a solução. Esse método é basicamente uma escolha aleatória em que existe a probabilidade de um indivíduo com má aptidão ser escolhido. Essa premissa tem a intenção de aumentar as chances de escapar de ótimos locais. O algoritmo é avaliado em apenas duas funções de *benchmark*.

A decomposição de problemas de otimização de maneira estática é apontada

por [Li e Yao, 2011] como um dos motivos para que algoritmos encontrem dificuldades para tratar problemas não separáveis. Para contornar esta limitação, os autores apresentam o algoritmo *Cooperative Coevolving PSO* que utiliza o conceito de “*random grouping*”. Ao dividir o problema em  $k$  subpopulações, são selecionadas partes do espaço de busca de modo aleatório. Outro conceito incorporado ao algoritmo é o “*adaptive weighting*” que favorece o ajuste fino das soluções incorporando um peso a cada dimensão do problema. É tomado o cuidado para que o peso aplicado a cada dimensão do problema não o faça extrapolar os limites da função. O algoritmo é executado em problemas com 30 dimensões e 6 enxames, com 500 dimensões e 5 enxames e com 1000 dimensões e 10 enxames. Os resultados foram comparados a variações do CPSO [Rakitienskaia e Engelbrecht, 2008] e o algoritmo apresentou melhor desempenho em todas as funções, exceto em uma (*Quadratic Function*). O número de partículas por enxame foi definido em 20 e em algumas funções o melhor valor alcançado não se aproximou do ótimo global.

O problema conhecido como “*curse of dimensionality*” é abordado por Parsopoulos [Parsopoulos, 2009]. O autor considera que a probabilidade de encontrar soluções dentro de um tempo razoável diminui à medida que a dimensão do problema aumenta e para contornar esta deficiência, propõe o COMPSO (do inglês, *Cooperative Micro-Particle Swarm Optimization*). O algoritmo divide cada problema em  $K$  vetores de três dimensões com 5 partículas cada. Com poucas partículas, a chance de estagnação é grande. Para contornar este problema, o algoritmo reinicializa independentemente cada enxame, mantendo as melhores posições caso não haja melhora na otimização. O algoritmo foi comparado ao algoritmo PSO original e apresentou melhores resultados. Porém, sabe-se que o PSO original não é uma boa solução para problemas com muitas dimensões. O trabalho também não explora topologias alternativas, usando somente a topologia em forma de anel.

Uma abordagem considerando o dinamismo de certos problemas de otimização é realizada por Xiangwei e Hong [Xiangwei e Hong, 2011] que observam que o algoritmo PSO é inadequado para funções em que o ótimo global modifica-se ao longo do

tempo. Esta deficiência é atribuída ao fato do algoritmo PSO não se adaptar às mudanças do ambiente. Para contrapor esta limitação, os autores propõem o DTMPSO (do inglês, *Different Topology Multi-swarm PSO*). São criados dois enxames - chamados de *G-swarm* e *L-Swarm* - em que o primeiro é totalmente conectado e o segundo é totalmente isolado. O *G-swarm* possui uma melhor partícula (*gbest*) que influencia todo o enxame, enquanto que no *L-swarm*, cada partícula é influenciada apenas por sua melhor posição (*pbest*) encontrada até o momento. Após um período fixo, as melhores partículas são trocadas entre os dois enxames. Também é utilizado um operador de mutação nas partículas para evitar a estagnação. O intuito é combinar a rápida convergência do *G-swarm* com a preservação da diversidade do *L-swarm* para a resolução de problemas dinâmicos. Os autores testam o algoritmo em funções de *benchmark* e obtêm bons resultados, todavia o fazem em problemas com apenas dez dimensões e não exploram um aspecto importante: a avaliação dos resultados com diferentes períodos de troca de partículas.

Vanneschi et al. [Vanneschi et al., 2010] introduzem quatro algoritmos para a resolução de problemas de otimização, todos com dez enxames e dez partículas por enxame. Um deles é uma implementação híbrida entre PSO e GA, em que após um período definido são aplicados operadores de mutação e *crossover*. O segundo algoritmo é uma variação do primeiro e inclui um componente de repulsa, em que, basicamente, as partículas são atraídas pelas melhores posições de seu próprio enxame e afastadas das melhores posições das outras partículas. Um terceiro algoritmo, chamado de MSPSO (do inglês, *Multi-swarm PSO*) usa um conjunto de enxames que executam o processo de otimização independentemente. Esse conjunto está conectado através de uma topologia em forma de anel e a cada dez iterações ocorre uma interação entre dois enxames: um enxame envia 20% de suas melhores partículas para o enxame vizinho. Este, por sua vez, substitui 20% de suas piores partículas pelas partículas recebidas. O quarto algoritmo apresentado pelo autores é uma variação do algoritmo anterior, que inclui o componente de repulsa previamente comentado. Tais algoritmos são testados em uma gama de funções de *benchmark*, variando

das consideradas mais simples às mais complexas, mostrando bons resultados para algumas e resultados não tão bons para outras. Os autores inclusive propõem novas funções de *benchmark* com complexidade considerada intermediária. Apesar do trabalho propor novas funções e utilizar múltiplos enxames, ele falha ao abordar funções com apenas 20 dimensões, ater-se a apenas uma topologia (no caso, a topologia em forma de anel) e não analisar o impacto da variação do período de troca de partículas no desempenho.

### 3.4.2 Taxonomia para identificar e classificar as estratégias de otimização cooperativa

Como perceptível, dentro da categoria de PSO baseados em múltiplos enxames, inúmeras possibilidades podem ser observadas. Considerando esta diversidade, El-Abd e Kamel [El-Abd e Kamel, 2008] propuseram uma taxonomia para identificar e classificar as estratégias de otimização baseadas na otimização cooperativa por enxame de partículas. Segundo os autores, ao criar uma solução cooperativa baseada em PSO, deve-se observar vários aspectos de modo a se obter melhores desempenhos. Tais aspectos foram condensados na forma de quatro questões:

1. **Qual informação trocar?** Em otimizações com múltiplos enxames, as informações mais comuns envolvem o *gbest*, um conjunto de partículas e líderes. Entretanto, o *gbest* de cada enxame pode representar informações diferentes. Por exemplo, o CPSO [Rakitienskaia e Engelbrecht, 2008] baseia-se na divisão do espaço de busca (vetor solução) em subespaços (vetores menores) onde cada subespaço é otimizado usando um enxame diferente. O vetor solução global é construído usando as soluções encontradas pela melhor partícula de cada enxame. Esta abordagem foi introduzida originalmente usando algoritmos genéticos [Potter e Jong, 1994]. Por outro lado, o *Island Model* consiste em um grupo de processos de otimização em paralelo (ilhas) que ocasionalmente envia indivíduos de um processo para outro para ajudar a propagar áreas recém

descobertas do espaço de busca consideradas promissoras. Todos os *gbest* representam o mesmo tipo de informação. Esta abordagem tem sido amplamente aplicada em algoritmos paralelos genéticos [Tomassini e Vanneschi, 2009]. Todavia, nos algoritmos genéticos, geralmente o número de indivíduos que migram de uma ilha para outra é maior que um. A Figura 3.6 ilustra esses modelos.

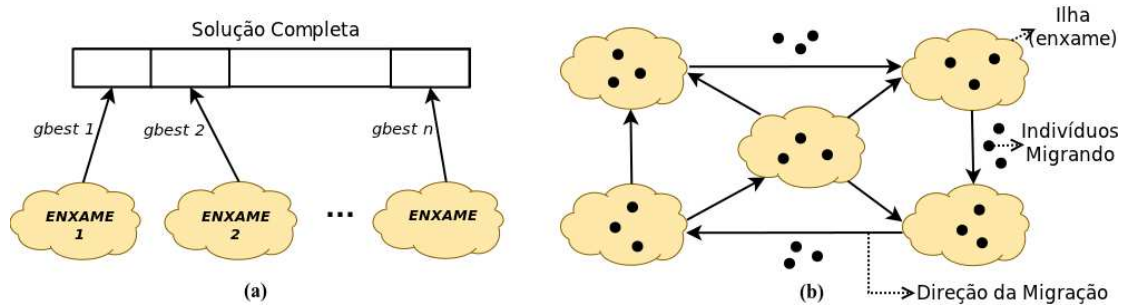


Figura 3.6: Duas abordagens para o compartilhamento de informações: (a) Modelo Co-evolutivo; (b) *Island Model*.

2. **Quando trocar a informação?** Essa questão envolve a estratégia de comunicação. Basicamente, existem duas estratégias: *síncrona* e *assíncrona*. O modelo de comunicação síncrono determina que informações são trocadas em um intervalo fixo de iterações. O modelo de comunicação assíncrono considera a ocorrência de um evento, como a estagnação do resultado após um número especificado de iterações.
3. **Como compartilhar a informação?** Esse aspecto está associado aos membros envolvidos na comunicação, que pode ocorrer apenas entre vizinhos ou entre todos os enxames. Especificamente nesta questão, é possível observar a relação entre os Sistemas Distribuídos e o PSO Cooperativo. Partindo dessa inferência, abordamos na Subseção 3.4.3 as **Estratégias para a distribuição de enxames** e seus aspectos que enquadram-se na comunicação cooperativa entre enxames e que ajudam a responder “Como compartilhar a informação”;
4. **O que fazer com a informação trocada?** Muitas ações estão envolvidas, como atualizar velocidades, substituir partículas ou reinicializar enxame.

As combinações de respostas associadas às questões propostas por [El-Abd e Kamel, 2008] podem gerar inúmeras possibilidades para configurar a Otimização por Enxame de Partículas com múltiplos enxames. Na Seção 5.1 do Capítulo 5 ponderamos algumas possibilidades e propomos uma abordagem cooperativa para a resolução de uma classe de problemas de otimização. Os resultados obtidos são apresentados, incluindo a discussão dos mesmos. As questões envolvidas na otimização com múltiplos enxames são consideradas e parâmetros são definidos. Tais experimentos envolvem a resolução de problemas de otimização com um objetivo e fundamentam a contribuição basilar dessa tese de doutorado: a resolução de problemas com vários objetivos considerando o uso de múltiplos enxames na otimização por enxame de partículas.

### 3.4.3 Estratégias para a distribuição de enxames

Como previamente exposto, a questão do compartilhamento de informações remete à *cooperação* entre enxames. Uma área da ciência da computação particularmente dedicada ao estudo de entidades independentes que cooperam é chamada de *Sistemas Distribuídos* (SD).

Segundo Kshemkalyani e Singhal [Kshemkalyani e Singhal, 2011], um SD é uma coleção de entidades independentes que cooperam para resolver um problema. Curiosamente, ao expor que sistemas distribuídos existem desde o “início do universo”, os autores valem-se de uma assertiva bioinspirada: “*From a school of fish to a flock of birds and entire ecosystems of microorganisms, there is communication among mobile intelligent agents in nature*”<sup>2</sup>.

Mullender [Mullender, 1993] define um SD como um sistema com várias entidades de processamento e vários dispositivos de armazenamento conectados entre si por uma rede. O autor destaca que outras definições podem ser apresentadas, porém propor uma em definitivo é arriscado, pois uma característica importante hoje ou que

---

<sup>2</sup>Em uma tradução livre: “desde um cardume de peixes a um bando de pássaros e completos ecossistemas de microorganismos, existe comunicação entre os agentes inteligentes móveis na natureza”.

eventualmente seja importante no futuro pode ser omitida.

Independente da definição, um SD é fundamentado pela *cooperação* entre seus membros. Assim, é necessário que os elementos de um SD comuniquem-se entre si. Normalmente, a comunicação em um SD ocorre através de troca de mensagens em uma rede, ou no acesso a memória compartilhada e realizar a comunicação de forma eficiente é um requisito para a construção de um SD tendo em vista propriedades desejadas, como a escalabilidade [Mullender, 1993]. Um MSPSO pode usar diferentes estratégias de cooperação entre as subpopulações. Elas são condicionadas à maneira como as informações são disseminadas, ou seja, às estratégias de comunicação empregadas. A seguir, são descritas topologias e estratégias de comunicação consideradas nesta tese.

## Topologias

A relação de interconexão entre os elementos de um SD é definida pela topologia de rede. A topologia pode ser vista como um grafo em que vértices representam os elementos e as arestas representam os elos entre esses elementos. A *topologia física* representa o meio de conexão dos dispositivos de redes e a forma como a estrutura física está conectada. A *topologia lógica* geralmente é definida no contexto de uma aplicação e determina a maneira como os dados são transmitidos através da rede a partir de um elemento para outro, considerando elos ou canais lógicos existentes. Doval e O'Mahony [Doval e O'Mahony, 2003] analisam a eficiência das *overlay networks* (redes de sobreposição), que estabelecem uma topologia virtual no topo de uma topologia física. Os autores concluem que modificações na topologia virtual para estabelecer topologias alternativas exercem forte influência na cooperação entre os elementos da rede.

Considerando os MSPSOs, a definição da topologia é fundamental para definir como os enxames estão conectados e, conseqüentemente, como eles cooperam entre si. As topologias de vizinhança definidas na Seção 3.3 do Capítulo 3 abordam a conexão entre partículas de um mesmo enxame, porém tais topologias também po-

dem ser empregadas para conectar subpopulações distintas entre si. A seguir, apresentamos três topologias adicionais que são comuns em sistemas distribuídos e que podem ser utilizadas em PSO com múltiplos enxames: anel, hipercubo e as topologias dinâmicas.

**Anel** (*ring*): topologia caracterizada como um caminho de transmissão que forma um ciclo lógico, sem um final definido. Como exposto na Seção 3.3, um nodo desta topologia possui dois vizinhos imediatos. Do ponto de vista das interconexões entre seus vértices, a transmissão pode ocorrer de maneira unidirecional ou bidirecional.

**Dinâmica**: são topologias que modificam-se ao longo do tempo. Assim, a conectividade entre os vértices pode ser aumentada ou reduzida à medida que o tempo passa. A Figura 3.7 ilustra um exemplo de topologia dinâmica que, ao longo do tempo, varia de uma topologia de grafo completo para uma topologia em anel [Montes de Oca et al., 2009]. A topologia dinâmica pode ser representada como um grafo  $G_t = (V_t, E_t)$ , o qual em um instante de tempo  $t$ ,  $V_t$  representa o conjunto de vértices do grafo naquele instante de tempo e  $E_t$  representa os enlaces entre os vértices de  $V_t$  naquele instante de tempo. Dados  $t_1$  e  $t_2$ , tal que  $t_1 \neq t_2$ ,  $G_{t_1}$  pode ser diferente de  $G_{t_2}$ .

**Hipercubo** (*hypercube*): o hipercubo (também chamado de *binary n-cube* ou *boolean n-cube*) consiste de uma rede *n-cubo* binária com  $2^n$  nodos, cada um dos quais interligado com  $n$  vizinhos [Saad e Schultz, 1988]. Conforme ilustrado na Figura 3.8, um *3-cubo* pode ser representado como um cubo com três dimensões, onde os vértices são os  $2^3 = 8$  nodos deste hipercubo. Um hipercubo é um grafo não direcionado consistindo de  $k = 2^n$  vértices identificados pelos  $2^n$  números binários entre 0 a  $2^n - 1$ . Uma aresta entre dois vértices é estabelecida se e somente se seus identificadores binários diferem em apenas um *bit*. Suas principais propriedades são [Saad e Schultz, 1988, Bona et al., 2008]: **a)** é uma estrutura recursiva. Para dois  $(n - 1)$ -cubos idênticos, cujos vértices também são numerados de 0 a  $2^{n-1} - 1$ , ao unir cada vértice do primeiro  $(n - 1)$ -cubo ao vértice do segundo com os mesmos identificadores, obtém-se um *n-cubo*; **b)** há  $n$  maneiras diferentes de se dividir um *n-cubo* em dois  $(n - 1)$ -cubos; **c)** um *n-cubo* não possui ciclos de comprimento ímpar; **d)** sendo o diâmetro de um



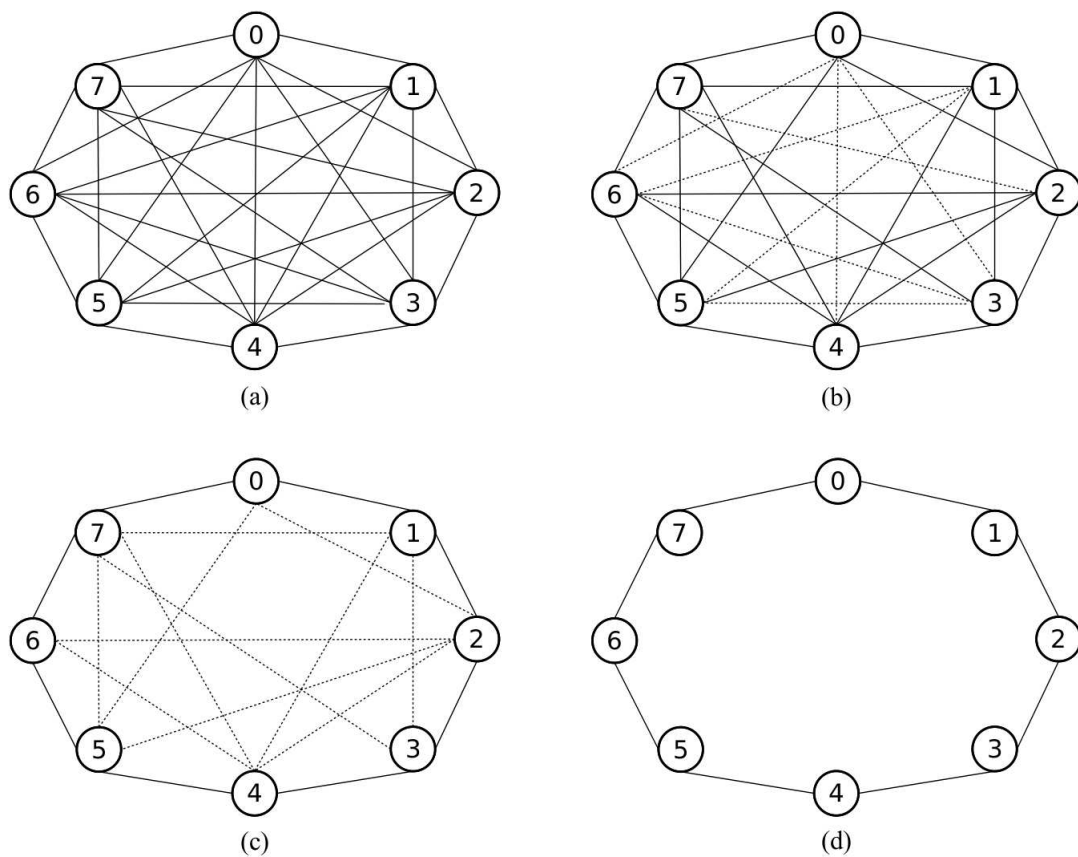


Figura 3.7: Exemplo de topologia dinâmica: inicialmente um grafo completo (a), que diminui sua conectividade à medida que o tempo passa (b) e (c), tornando-se uma topologia em anel (d).

grafo a maior distância entre quaisquer dois de seus vértices, seu valor para o  $n$ -cubo ( $2^n$  vértices) é  $n$ ; **f)** a menor distância entre dois vértices de um hipercubo é igual ao número de *bits* em que tais vértices diferem. O hipercubo é uma estrutura que escala bem, pois possui diversas propriedades logarítmicas, entre elas o grau e a maior distância que separa dois nodos quaisquer.

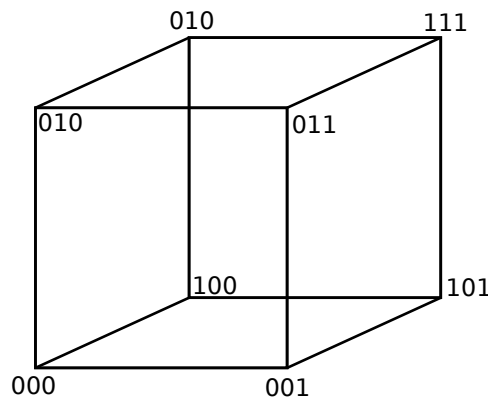


Figura 3.8: Visualização de um hipercubo 3-cubo.

## Estratégias de comunicação

A disseminação das informações é realizada pela estratégia adotada para propagá-las. Basicamente, almeja-se um equilíbrio entre tráfego (quantidade de mensagens transmitidas) e cobertura (número de nodos que recebem a mensagem). No contexto do MSPSO, a alta conectividade entre os enxames implica em um rápido compartilhamento de melhores partículas, que pode levar à estagnação em ótimos locais. Todavia, o oposto pode fazer com que enxames fiquem isolados e não aproveitem as boas posições encontradas por partículas de outras subpopulações. As principais estratégias de comunicação utilizadas neste trabalho são descritas a seguir: *broadcast* e *gossip*.

**Broadcast:** também chamada de difusão, é uma técnica simples em que uma mensagem é enviada de um nodo para todos os outros nodos. Nesse caso, há apenas um remetente, mas as informações são enviadas para todos os receptores conectados, apresentando ótima cobertura [Hedetniemi et al., 1988]. A transmissão por

difusão permite de forma trivial o envio de uma mesma mensagem para todos os envolvidos na comunicação. A Figura 3.9(a) ilustra uma estratégia de comunicação *broadcast*, em que o nodo 0 envia uma mensagem para todos os demais. Adicionalmente, são ilustradas as estratégias de comunicação *unicast* e *multicast* (difusão seletiva). Na transmissão *unicast*, uma mensagem é enviada de um nodo para um destino específico: existe apenas um remetente e um receptor, nodos 0 e 2, respectivamente (Figura 3.9(b)). Na transmissão *multicast* (Figura 3.9(c)), uma mensagem é enviada pelo nodo 1 para um grupo específico, consistindo dos nodos 3, 5 e 7.

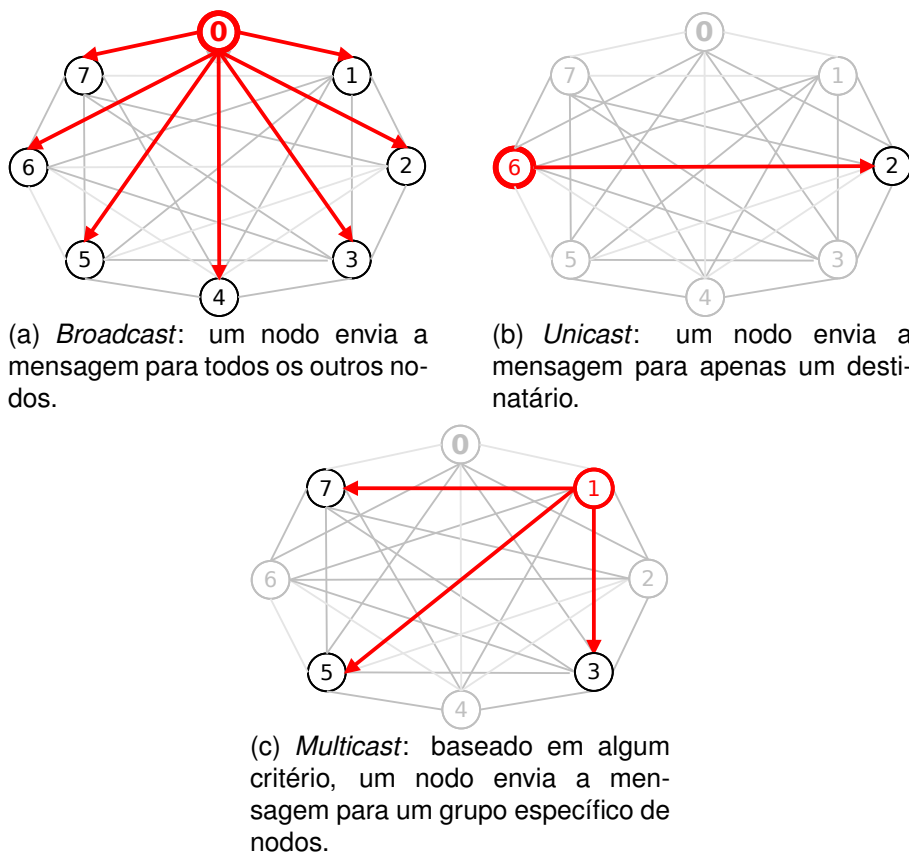


Figura 3.9: Estratégias básicas de comunicação: remetentes representados por vértices em vermelho e destinatários representados por vértices em preto.

**Gossip:** apesar dos pontos positivos da estratégia de comunicação *broadcast*, como a taxa de cobertura e simplicidade [Liu et al., 2008], esta técnica tem como principal desvantagem a sobrecarga que impõe à rede e o consumo demasiado de recursos, devido ao potencialmente grande número de mensagens utilizadas [Liang et al., 2007]. Para amenizar tais desvantagens é possível diminuir o número de saltos

que a transmissão realizará entre os nodos ou reduzir o número de nodos que receberão a consulta. Entretanto, como observado por Yadong et al. [Yadong et al., 2010] estas alternativas podem comprometer a abrangência de cobertura. Várias iniciativas são propostas para obter uma boa cobertura, reduzindo o número de mensagens retransmitidas. Uma alternativa é a *disseminação epidêmica* [Eugster et al., 2004, Van Renesse et al., 1998, Biazzi e Montresor, 2010], também conhecida por *gossip*, cujo protocolo foi inicialmente proposto por Baker et al [Baker et al., 1972]. O algoritmo epidêmico é executado por todos os nodos. Basicamente, um nodo encaminha mensagens de um *buffer*  $b$  em intervalos de tempo  $t$  para outros nodos (um ou mais) escolhidos aleatoriamente. O número de membros escolhidos é limitado a um tamanho  $f$ , chamando de *fan-out* da disseminação [Eugster et al., 2004]. Existem muitas variações de algoritmos epidêmicos que são tipicamente distinguidos entre si pelos valores de  $b$ ,  $t$  e  $f$ . A cobertura de entrega da informação é influenciada por esses parâmetros.

## CAPÍTULO 4

### A OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS APLICADA À RESOLUÇÃO DE PROBLEMAS COM MAIS DE UM OBJETIVO

O PSO estabeleceu-se como um eficiente algoritmo de otimização e é possível encontrar uma variedade de aplicações da vida real em que ele foi utilizado com sucesso [Parasopoulos e Vrahatis, 2002]. Ao longo do tempo, surgiram diversas propostas para estender o PSO para resolver problemas multiobjetivos [Coello Coello e Lechuga, 2002, Hu e Eberhart, 2002, Mostaghim e Teich, 2003, Padhye et al., 2009, Chakraborty et al., 2010, de Carvalho e Pozo, 2011a, de Carvalho e Pozo, 2011b, de Britto et al., 2013]. Neste capítulo inicialmente são abordados conceitos relacionados à adaptação do algoritmo PSO para resolver problemas de otimização com mais de um objetivo. A seguir, aspectos associados à seleção de líderes e ao gerenciamento de soluções não dominadas são tratados. Finalmente, os principais algoritmos desenvolvidos na área são discutidos.

#### 4.1 *Multiple objectives PSO - MOPSO*

Como dito no Capítulo 3, no PSO, o movimento de cada partícula é condicionado por seu conhecimento individual ( $pbest$ ) e pelo conhecimento coletivo ( $gbest$ ). Algoritmos baseados em PSO para resolver problemas com mais de um objetivo são chamados de *Multi-objective Particle Swarm Optimization* (MOPSO). Segundo Padhye [Padhye, 2009], o grande desafio ao se estender o PSO para problemas multiobjetivos é decifrar a noção de líder. Em um cenário multiobjetivo existe um conjunto de  $[p][g]bests$  ao invés de um único ótimo local/global. Assim, a efetividade de um MOPSO está particularmente sujeita à estratégia utilizada para a seleção de líderes locais e globais. Esses líderes são mantidos em um repositório (arquivo) externo que armazena o con-

**Algoritmo 4:** Pseudo-código do MOPSO.**Entrada:** Problema de Otimização**Saída:** Soluções não dominadas do arquivo externo  $A$ 


---

```

1: para cada partícula  $i$ 
2:   inicialize  $\vec{x}_i, \vec{v}_i, pbest_i \leftarrow \vec{x}_i$ ;
3:  $A \leftarrow \emptyset$ 
4: para cada partícula  $i$ 
5:   calcule o valor de aptidão de  $\vec{x}_i$ 
6:   baseado no conceito de dominância de Pareto, armazene as partículas não
     dominadas no repositório  $A$ 
7: fim-para
8: enquanto número máximo de iterações não atingido
9:   para cada partícula  $i$ 
10:    selecione o líder  $rep_{lider}$  a partir do repositório  $A$ 
11:    atualize a velocidade  $\vec{v}_i \leftarrow \vec{v}_i + rand() \cdot (pbest_i - \vec{x}_i) + rand() \cdot (rep_{lider} - \vec{x}_i)$ 
12:    atualize a posição  $\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$ 
13:    atualize o repositório  $A$  usando um arquivador
14:    se  $(f(\vec{x}_i) \prec f(pbest_i))$ 
15:      então  $pbest_i \leftarrow \vec{x}_i$ 
16:    fim-se
17:    se  $(f(\vec{x}_i) \text{ e } f(pbest_i) \text{ possuem o mesmo grau de dominância})$ 
18:      então  $pbest_i \leftarrow pbest_i$  ou  $pbest_i \leftarrow \vec{x}_i$  (decisão aleatória)
19:    fim-se
20:  fim-para
21: fim-enquanto.

```

---

junto de soluções não dominadas encontradas até o momento pelo processo de otimização. Geralmente, este repositório é limitado a um tamanho e um **arquivador** deve controlar este limite, pois o número de soluções não dominadas pode crescer rapidamente. Um arquivador deve usar diferentes estratégias para garantir convergência e diversidade à busca [López-Ibáñez et al., 2011]. O pseudocódigo do MOPSO genérico é apresentado no Algoritmo 4.

Conforme apresentado no Algoritmo 4, o primeiro passo consiste em iniciar, aleatoriamente os valores de cada partícula e sua velocidade dentro dos limites definidos pelo problema. Inicialmente, a melhor posição de cada partícula é seu próprio valor e o arquivo externo ( $A$ ) com as soluções não dominadas está vazio. Após, um laço é executado para avaliar (em relação a cada objetivo) a aptidão de cada partícula. Baseado na teoria de dominância de Pareto, as soluções não dominadas são incluídas no arquivo externo  $A$ . A seguir, o seguinte processo é repetido até que um limite de

iterações seja atingido:

- para cada partícula  $i$ , um líder  $rep_{l\acute{i}der}^{\vec{}}$  é selecionado do repositório  $A$  e a velocidade  $\vec{v}_i$  da partícula  $i$  é atualizada. O líder é equivalente ao  $gbest^{\vec{}}$  da equação 3.2;
- baseado no cálculo da velocidade  $\vec{v}_i$ , uma nova posição para  $\vec{x}_i$  é determinada;
- utilizando conceitos de dominância de Pareto, a nova posição de  $\vec{x}_i$  é comparada às posições não dominadas presentes no arquivo externo  $A$ . Cabe ao *arquivador* determinar quais partículas serão mantidas em  $A$ ;
- finalmente, a melhor posição individual ( $pbest$ ) da partícula  $i$  é atualizada. Aplica-se o conceito de dominância de Pareto entre  $pbest_i^{\vec{}}$  e  $\vec{x}_i$  e, se a posição corrente  $\vec{x}_i$  domina  $pbest_i^{\vec{}}$ , então  $pbest_i^{\vec{}}$  é substituída por  $\vec{x}_i$ . Caso não haja relação de dominância entre as posições corrente/melhor individual de  $i$ , uma delas é selecionada aleatoriamente como um novo  $pbest_i^{\vec{}}$ . Naturalmente, nada é feito se  $pbest_i^{\vec{}}$  domina  $\vec{x}_i$ . Técnicas mais elaboradas de seleção de líderes podem ser empregadas para definição do  $pbest_i^{\vec{}}$  [Mostaghim e Teich, 2003].

Quando o limite de iterações é atingido, o algoritmo gera como saída o conjunto de soluções não dominadas que estão armazenadas no arquivo externo  $A$ .

Como observado no Algoritmo 4, os componentes do MOPSO são semelhantes aos do PSO. As principais diferenças são duas: 1) a definição da melhor posição do enxame, que é escolhida como um líder global para a partícula  $i$ , a partir do repositório de soluções não dominadas; e 2) o método de arquivamento utilizado pelo MOPSO para gerenciar o repositório com as soluções não dominadas. A primeira diferença está associada, principalmente, à cooperação entre as partículas. A segunda diz respeito à manutenção do repositório para tentar melhorar a distribuição das soluções ou para manter o número de soluções não dominadas dentro da capacidade do repositório. A seguir, serão discutidos alguns dos principais métodos para seleção de líderes e arquivamento.

## 4.2 O desafio para determinar os líderes no MOPSO

A escolha da partícula  $rep_{lider}$  mantida no arquivo externo  $A$  do Algoritmo 4 é de significativo impacto no desempenho dos MOPSOs. Esta partícula atuará como um líder (melhor solução global) e influenciará no movimento das partículas do enxame. Consequentemente, a seleção de líderes determinará aspectos de convergência de diversidade do enxame como um todo [Mostaghim e Teich, 2003, Hu e Yen, 2013]. Um candidato pode ser selecionado do arquivo externo através de diferentes estratégias que influenciarão o desempenho do MOPSO. A seleção de líderes é foco de intensa pesquisa e as principais estratégias são destacadas a seguir:

- *Escolha aleatória*: é a estratégia mais simples e considerada a computacionalmente menos onerosa para seleção de líderes. A partícula líder é selecionada aleatoriamente do repositório e em situações onde existam regiões populosas este método pode levar a uma indesejada perda de diversidade.
- *Método Sigma* [Mostaghim e Teich, 2003]: o líder de cada partícula é o indivíduo mais próximo a uma linha traçada entre a partícula e a origem. É um método que apresenta rápida convergência, mas pode levar à estagnação das soluções em ótimos locais.
- *Weighted Sum (WSum)* [Branke e Mostaghim, 2006]: método que prioriza a diversidade das soluções ao associar um peso maior àqueles objetivos nos quais a partícula é melhor. Seja  $f_j(x_i)$  o  $j_{esimo}$  valor de aptidão da partícula  $i$ , para cada partícula candidata a líder da partícula  $i$ , a soma ponderada será calculada como  $F = \sum_j \frac{f_j(x_i)}{\sum_k f_k(x_i)} f_j(\vec{p}_i)$ . Em problemas de minimização, a partícula com a maior soma ponderada será a escolhida.
- *NWSum*: variação do método *WSum*. Novamente, a soma ponderada é calculada através da equação acima e o líder é selecionado baseado no menor valor obtido.



- *Seleção baseada em indicador (Indicator Based)*: este método emprega o *hypervolume* como indicador para seleção de líderes. O *hypervolume* é computado para todo o repositório em relação à partícula e o membro do arquivo que melhor contribua para o indicador é escolhido como líder [Hu e Yen, 2013].
- *Crowding Distance*: amplamente empregado pelos MOEAs para promover diversidade nos resultados, o uso do *Crowding Distance* em MOPSOs para selecionar líderes foi inicialmente adotado por Raquel e Naval [Raquel e Naval, 2005]. Esse método determina as regiões mais populosas do conjunto de soluções e evita selecionar pontos localizados nelas. O desempenho desta abordagem mostrou-se competitivo em termos de convergência e diversidade.
- *Maximin*: proposto por Li [Li, 2004] para selecionar líderes em MOPSO, este método calcula o valor *maximin* de  $u$  através de:  $f_{maximin} = \max_{j=1, \dots, m; u \neq v} \{ \min \{ f_i(\vec{u}) - f_i(\vec{v}) \} \}$ . Indivíduos com os menores valores *maximin* são preferidos. Para obter o valor *maximin* de todo o repositório, este método tem complexidade geral  $O(mP^2)$ , onde  $m$  é o número de objetivos e  $P$  é o tamanho da população. É um método que funciona bem para algoritmos com população elevada e favorece regiões centrais em fronteiras do tipo convexa e regiões extremas em fronteiras côncavas.
- *Diversity* [Branke e Mostaghim, 2006] como o nome sugere, é um método proposto para favorecer a diversidade. Para cada partícula, calcula-se a distância mínima entre ela e todas as partículas do repositório. A partícula com maior distância (a mais isolada) é selecionada como líder. Este método força a partícula a explorar regiões menos densas.
- *Opposite Method* [Castro et al., 2012]: método que também visa obter diversidade de soluções. Similarmente ao método *Diversity*, a ideia é mover as partículas para diferentes áreas do espaço de objetivos. Basicamente, dada uma partícula  $i$ , com o vetor objetivo  $f(\vec{x}_i)$ , o novo líder de  $i$  é a solução do repositório com um vetor objetivo em uma região oposta de  $f(\vec{x}_i)$ . Ou seja, a partícula es-

---

**Algoritmo 5:** Pseudo-código do Arquivador Preciso [López-Ibáñez et al., 2011].

---

**Entrada:**  $A^{t-1}$ ,  $\vec{x}$ ,  $N$  (Arquivo anterior, solução atual, tamanho máximo do arquivo)

**Saída:**  $A^t$  (Arquivo atualizado)

- 1: **se**  $\exists \vec{y} \in A^{t-1}$ , tal que  $f(\vec{x}) \prec f(\vec{y})$  **então**
  - 2:    $A^t \leftarrow A^{t-1} \cup \vec{x} / \{\forall \vec{y} \in A^{t-1}, f(\vec{x}) \prec f(\vec{y})\}$  **(1)**
  - 3: **senão se**  $\exists \vec{y} \in A^{t-1}$ , tal que  $f(\vec{y}) \prec f(\vec{x})$  **então**
  - 4:    $A^t \leftarrow A^{t-1}$  **(2)**
  - 5: **senão se**  $\nexists \vec{y} \in A^{t-1}$ , tal que  $f(\vec{y}) \prec f(\vec{x})$  ou  $f(\vec{x}) \prec f(\vec{y})$  **então**
  - 6:   **se**  $(|A| < N)$  **então**
  - 7:      $A^t \leftarrow A^{t-1} \cup \vec{x}$  **(3)**
  - 8:   **senão**  $A^t \leftarrow \text{filtro}(A^{t-1}, \vec{x})$  **(4)**
- 

colhe um líder que tenha bons valores objetivo naqueles objetivos onde  $f(\vec{x}_i)$  não possua resultados tão bons.

É importante destacar que os métodos empregados para selecionar a melhor solução global podem ser adaptados para serem utilizados na seleção da melhor solução individual de cada partícula. Boas referências neste tema podem ser consultadas em [Padhye et al., 2009, Britto e Pozo, 2012].

### 4.3 Gerenciamento de soluções não dominadas na otimização multiobjetivo

O uso de repositórios de soluções não é exclusividade dos MOPSOs. Segundo Coello *et al.* [Coello et al., 2007], muitos MOEAs utilizam uma segunda população chamada de arquivo, repositório ou arquivo externo para armazenar soluções não dominadas encontradas através do processo evolutivo. A cada iteração considera-se adicionar uma solução ao arquivo utilizando estratégias que comparam esta solução com as soluções atualmente presentes no arquivo. De acordo com Lopes-Ibanez *et al.* [López-Ibáñez et al., 2011], a maioria dos arquivadores seguem o esquema apresentado no algoritmo 5 e pertencem à classe de “*arquivadores precisos*”.

Considerando que  $A$  é um arquivo com vetores não dominados, onde  $|A| \leq N$  ( $N$  = tamanho máximo do arquivo), um arquivador preciso apresentará as seguintes

propriedades (valores numéricos entre parênteses e destacados em negrito no algoritmo 5):

1. considerando o vetor  $\vec{x}$ , se  $\vec{x}$  domina alguma solução em  $A^{t-1}$ ,  $\vec{x}$  é adicionado em  $A^{t-1}$  e os pontos dominados por  $A^{t-1}$  são eliminados, obtendo-se  $A^t$ ;
2. se  $\vec{x}$  é dominado por alguma solução de  $A^{t-1}$ ,  $A^{t-1}$  não é alterado ( $A^{t-1}=A^t$ );
3. caso  $|A^{t-1}| < N$  e  $\vec{x}$  não domina nenhuma solução em  $A^{t-1}$  e não é dominado por nenhuma solução em  $A^{t-1}$ , então é  $\vec{x}$  adicionado a  $A^t$ ;
4. caso  $|A^{t-1}| = N$  e  $\vec{x}$  não domina nenhuma solução em  $A^{t-1}$  e não é dominado por nenhuma solução em  $A^{t-1}$ , então a função *filtro* é utilizada para determinar quais as  $N$  soluções permanecerão no conjunto  $A^{t-1} \cup \vec{x}$ .

Ainda de acordo com [López-Ibáñez et al., 2011], a maneira mais simples de lidar com a questão do arquivamento é implementar o *arquivador básico*, com a função *filtro* que simplesmente retorna  $A^{t-1}$ . Ou seja, sempre que o repositório estiver cheio, este arquivador admite apenas soluções que dominem outras no arquivo externo. Apesar de simples, este arquivador não garante a qualidade da aproximação, principalmente quando  $N$  for pequeno. Assim, técnicas mais elaboradas tornam-se necessárias. A seguir, algumas delas são apresentadas [López-Ibáñez et al., 2011, Britto e Pozo, 2012].

- *NSGA-II Archiver*: associa valores de seleção para cada ponto do repositório de arquivos baseado em sua distância de agrupamento (CD, *Crowding Distance*). O ponto com menor CD será removido, pois deduz-se que este se encontra em uma região da fronteira mais populosa;
- *SPEA2 Archiver*: utiliza a função de filtro baseada na menor distância do ponto e seu vizinho como uma medida de densidade. Possui propriedade similares ao arquivador do NSGA-II em termos de convergência e aproximação;

- *Adaptive Grid Archiving (AGA)*: utiliza células de *grid* sobre os pontos para estimar a densidade local. O novo ponto  $\vec{x}$  é adicionado ao arquivo e o arquivador reconstrói o *grid*, removendo aleatoriamente um ponto da célula mais povoada;
- *Hypervolume Archiver*: remove o ponto do arquivo que menos contribui para o *hypervolume*. É considerado um método que atinge bons resultados, porém devido a seu custo computacional, sua utilização torna-se impraticável à medida que  $N$  e a dimensão dos vetores aumentam;
- *Unbound Archiver*: baseado em um repositório de tamanho ilimitado, a função *filtro* retorna  $A^{t-1} \cup \vec{x}$ ;
- *Ideal Archiver*: aqui, a função *filtro* obtém o ponto ideal ( $z^*$ ) do conjunto  $A^{t-1} \cup \vec{x}$  e então calcula a distância Euclidiana de cada ponto no arquivo até o ponto ideal. Após, o ponto com maior distância é removido, mantendo os pontos do arquivo próximos ao ponto ideal [Britto e Pozo, 2012];
- *Adaptive  $\varepsilon$ -Approx Archiving ( $\varepsilon app$ )*: baseado no conceito de  $\varepsilon$ -dominância, esta técnica aceita um novo ( $\vec{x}$ ) apenas se este domina pela  $\varepsilon$ -dominância outro elemento do arquivo. Proposto por Laumanns *et al.* [Laumanns et al., 2002],  $\varepsilon$ -dominância é uma dominância de Pareto relaxada em que  $\varepsilon$  é um parâmetro que pode ser manipulado;
- *Adaptive  $\varepsilon$ -Pareto Archiving ( $\varepsilon aps$ )*: usa a ideia de que o espaço de objetivos pode ser discretizado via  $\varepsilon$  em classes esquivariantes chamadas de caixas. Neste conjunto de caixas de tamanho  $\varepsilon$ , apenas uma solução não dominada é mantida em cada caixa. Se dois ou mais pontos pertencem a uma mesma caixa, o ponto mais próximo ao índice da caixa é selecionado.

Técnicas de arquivamento e seleção de líderes são fundamentos essenciais para o desenvolvimento de algoritmos multiobjetivos baseados em PSO. A seguir, os principais MOPSOs, no contexto dessa tese, são revisados.

## 4.4 MOPSO - Estado da arte

A primeira iniciativa relevante para utilizar a otimização por enxame de partículas para resolver problemas multiobjetivos foi apresentada por Coello e Lechuga [Coello Coello e Lechuga, 2002]. A proposta, chamada apenas de MOPSO (do inglês, *Multi-Objective Particle Swarm Optimization*) utiliza a fronteira de Pareto para conduzir o “vôo” das partículas e mantém um repositório global com todas as soluções não dominadas. Para aumentar a diversidade, os autores dividem o espaço de busca de modo a criar “limites geográficos” para determinar a que região uma solução não dominada pertence no repositório. Quando uma partícula atualiza sua velocidade, um líder é escolhido aleatoriamente desse repositório. A cada ciclo, o repositório é atualizado: soluções dominadas são eliminadas e soluções não dominadas são incluídas. Como o espaço do repositório é limitado, ao atingir esse limite, o arquivador tenta manter as soluções pertencentes às regiões menos populosas.

Os autores utilizam três *benchmarks* e comparam com dois algoritmos evolutivos: NSGA II (do inglês, *Non-dominated Sorting Genetic Algorithm II*) e PAES (do inglês, *Pareto Archived Evolution Strategy*). Os resultados obtidos pelo MOPSO são melhores que os outros algoritmos, porém algumas questões ficam em aberto e merecem ser consideradas. Problemas com vários objetivos não são tratados e apenas duas métricas foram utilizadas. Uma delas, que considera o tempo de execução (medido em segundos), é muito subjetiva, pois o tempo é fortemente influenciado pelo escalonamento de processos do sistema operacional. Os resultados deveriam ser avaliados considerando métricas que pudessem avaliar mais apropriadamente a qualidade das soluções, mensurando a diversidade e convergência das mesmas.

Sobre o tamanho do repositório global, os autores expõem que esse parâmetro exerce influência na qualidade da fronteira de Pareto produzida, mas nenhum experimento é conduzido abordando esse aspecto. E, finalmente, considerando a adoção de múltiplos enxames para resolver problemas multiobjetivo e com vários objetivos, a gerência de tais repositórios mostra-se de grande importância.

O uso do PSO para resolver problemas multiobjetivo é apresentado por Hu e Eberhart [Hu e Eberhart, 2002] como uma escolha natural, pois ao fim do processo, como resultado, gera-se uma população de resultados. Tal população é adequada à característica de problemas com muitos objetivos que possuem como solução um conjunto de soluções e não apenas uma única solução. Os autores implementam um esquema para definir a vizinhança dos indivíduos que é determinada pela distância entre as partículas. Assim, é definido um ótimo local (*lbest*) em cada vizinhança, ao invés de existir um único *gbest* para todas as partículas. Consequentemente, surgem dois desafios adicionais: 1) definir a distância entre as partículas; e 2) determinar qual é a melhor partícula da vizinhança. A distância de uma partícula para as outras é calculada baseada no valor de aptidão das partículas em relação à função do primeiro objetivo. Em seguida, são escolhidas as  $m$  partículas mais próximas entre si ( $m$  determina o tamanho da vizinhança). A cada ciclo, as posições das partículas se alteram, assim a vizinhança de partículas é dinâmica. Em relação à melhor partícula da vizinhança (*lbest*), ela é escolhida em função do melhor valor de aptidão, considerando o segundo objetivo. Ou seja, a cada execução, um objetivo é fixado e o outro é otimizado. Os autores comparam os resultados com soluções baseadas em GA e concluem que os resultados obtidos pelo PSO são melhores. Todavia, a comparação é baseada na verificação visual dos gráficos traçados em função dos resultados obtidos. É uma análise subjetiva, não endossada por alguma métrica. O conjunto de testes considerou apenas dois objetivos para cada problema e os dois objetivos não foram simultaneamente otimizados a cada rodada.

Um trabalho apresentado por Mostaghim e Teich [Mostaghim e Teich, 2003] considera a seleção de líderes na otimização multiobjetivo utilizando o algoritmo PSO como um aspecto de grande impacto na diversidade e convergência das soluções. É enfatizado que problemas multiobjetivo resolvidos pelo PSO requerem a definição de qual partícula será utilizada como líder para ser seguida por outros indivíduos. Os autores propõem um método chamado ***sigma*** em que é associado um valor sigma para cada partícula e para cada indivíduo mantido no repositório com o conjunto de

soluções não dominadas. Esse valor é obtido através de uma equação calculada em função da posição da partícula  $i$  no espaço de soluções dada por sua coordenada ( $f1_i$  e  $f2_i$  para dois objetivos e  $f1_i$ ,  $f2_i$  e  $f3_i$  para três objetivos). Quando uma partícula é atualizada, a escolha do indivíduo pertencente ao repositório e que será definido líder ( $gbest$ ) é condicionada pela menor distância entre a partícula e a solução não dominada do repositório. Esse valor compreende a distância Euclidiana entre a partícula e o líder. Os autores adicionam o que chamam de fator de turbulência que funciona de maneira análoga ao operador de mutação dos algoritmos genéticos. Esse fator serve para fazer uma mudança abrupta na posição da partícula e é dependente de um número aleatório entre 0 e 1.

Os experimentos são validados executando quatro *benchmarks* para dois e três objetivos. Os autores comparam os resultados com outras três soluções, baseadas em PSO e GA e concluem que os resultados obtidos para dois objetivos são melhores que as outras soluções e que a convergência obtida, considerando problemas com três objetivos é pior apenas que a solução baseada em GA. Porém, apenas problemas com dois e três objetivos são considerados. Além disso, a dimensão dos problemas com dois objetivos é de tamanho igual a 30 e para problemas com três objetivos, a dimensão do problema é de apenas três. Outro aspecto a ser considerado é que a validação dos resultados não contempla o uso de um conjunto amplo de métricas.

No algoritmo PSO, a velocidade da partícula é atualizada considerando que existe uma tendência dela manter essa velocidade (momento) e que ela pode ser influenciada em direção à melhor posição encontrada pela própria partícula ( $pbest$ ) e em direção à melhor posição encontrada por todas as partículas ( $gbest$ ). No contexto desse trabalho, as influências exercidas pelas melhores partículas individual e global são tratadas como *acelerações*. Em [Daneshyari e Yen, 2008], Daneshyari e Yen observam que a maioria das soluções baseadas em PSO para resolver problemas com mais de um objetivo atualizam a velocidade da partícula considerando apenas parâmetros fixos para o momento e para as acelerações. Enfatizam que, até então, nenhum trabalho propôs a adaptação dinâmica desses fatores para cada partícula.

Os autores implementam um MOPSO em que cada partícula possui seus próprios parâmetros para o momento e aceleração (individual e local) para se aproximar da fronteira de Pareto. Para isso, os autores valem-se do conceito de “cultura” em que define-se que o armazenamento do conhecimento não depende de quem os gerou e que ele pode ser acessado por todos os membros da sociedade. Na prática, além da população tradicional do PSO é criada uma base de dados que mantém os melhores indivíduos e informações sobre os menores e os maiores valores referentes às posições e às velocidades desses indivíduos (conhecimento cultural). A cada iteração, através de diversas equações, as acelerações local e global, além do momento são calculados para cada partícula em função desta base de dados. O *gbest* também é obtido dessa base que é atualizada a cada ciclo. Apenas um problema é avaliado, e além da visualização de alguns gráficos a única métrica de desempenho utilizada é o *hypervolume*. Por fim, há de se considerar que o custo computacional para gerar o “conhecimento cultural”, que envolve a adoção de muitas equações no processo de atualização e seleção não pode ser desconsiderado.

Segundo Padhye *et al.* [Padhye et al., 2009], o maior desafio dos MOPSO está relacionado à escolha de líderes envolvendo melhores partículas individual (*pbest*) e global (*gbest*). Segundo os autores, esta é a principal característica de qualquer MOPSO. São comparados vários métodos de seleção de líderes/melhores partículas. Além do repositório com as soluções globais não dominadas, é analisada a influência nos resultados pela adoção de repositórios individuais para cada partícula. Esse repositório individual mantém um histórico das melhores soluções não dominadas de cada partícula. Assim, além de escolher um líder (*gbest*), cada partícula escolhe um líder a partir de seu próprio histórico (*pbest*). A maioria das implementações considera que o *pbest* é a mais recente solução não dominada produzida pela própria partícula no processo iterativo. Os autores de [Padhye et al., 2009] propõem então o que chamam de “*velocity trigger*”. Ele tem um funcionamento semelhante ao fator de turbulência, que, por sua vez, serve como um operador de mutação dos GA. A ideia consiste em fixar uma quantidade de iterações em que a velocidade mantém-se abaixo



de um limite pré-definido. Deduz-se que nessas condições, não há progresso na otimização e provavelmente a partícula está estagnada em um ótimo local. Assim, novos valores para a velocidade são aleatoriamente gerados e espera-se que a partícula fuja do ótimo local. Por fim, é investigado como o desempenho é afetado pelo ajuste dos parâmetros dos diversos métodos e conclui-se que alguns métodos são mais sensíveis a esses ajustes que outros métodos. Os experimentos são executados em problemas com 30 dimensões e o *hypervolume* é utilizado como métrica. Apesar de mecanismos complexos de escolha de líderes, envolvendo muitas equações matemáticas, em que cada método apresenta pontos positivos e negativos em relação à convergência, diversidade e custo computacional, os autores afirmam que a escolha aleatória apresenta um bom equilíbrio: é a que requer menos recursos computacionais e, mesmo assim, gera bons valores de *hypervolume*. O trabalho é uma excelente referência que aborda métodos para selecionar líderes, porém é sabido que uma única métrica de desempenho não é adequada para avaliar a qualidade dos resultados em problemas multiobjetivo.

Muitos trabalhos publicados avaliam a estabilidade e a propriedade de convergência de algoritmos PSO para resolver problemas com um objetivo. Porém, considerando algoritmos PSO multiobjetivo essa avaliação é inexistente. Essa é a constatação de Chakraborty et al. [Chakraborty et al., 2010] e que motivou os autores a desenvolverem essa investigação. Os autores consideram a adoção de diferentes parâmetros de momento e acelerações e provam matematicamente que o enxame não convergirá para fronteira de Pareto se a soma das constantes de aceleração dividido por dois for maior que o valor do momento mais um multiplicado por 2, ou seja, considerando a equação 3.2 do Capítulo 3:  $\frac{C1+C2}{2} \geq (\omega + 1) \cdot 2$ . Os autores utilizaram o MOPSO apresentado por Coello e Lechuga [Coello Coello e Lechuga, 2002] e executaram dois problemas com dois objetivos, cada. Dois conjuntos de parâmetros para o momento e acelerações foram utilizados: um respeitando a prova matemática e outro, não. Gráficos são apresentados e os autores provam que a teoria apresentada é válida na avaliação prática. O trabalho tem o mérito de mostrar como tais parâmetros influenciam

o desempenho do MOPSO, porém é necessário avaliá-los em outros *benchmarks*, e principalmente, essa influência em problemas com mais de dois objetivos, potencialmente com vários objetivos.

Ao apresentar a revisão bibliográfica sobre os MOPSO disponíveis, Zhongkai *et al.* [Zhongkai et al., 2010] destacam três principais deficiências: **1)** custo computacional para escolher líderes; **2)** tendência de estagnar em regiões da fronteira devido à fraca habilidade de busca global; e **3)** capacidade de lidar com apenas dois objetivos. Como consequência, os MOPSO propostos apresentam dificuldade em gerar soluções com boa diversidade. Assim, os autores apresentam o DSMOPSO (do inglês, *Distance Sorting-based Multi-Objective Particule Swarm Optimization*). Nesse algoritmo, a manutenção do repositório de soluções não dominadas considera a distância de agrupamento (*Crowding Distance*, CD [Srinivas e Deb, 1994]). A função CD define o perímetro do cuboide criado a partir de um ponto, tendo como vértices, os vizinhos desse ponto. Uma vez que tais valores são ordenados, os indivíduos com maior valor de CD são escolhidos como líderes, ou seja, partículas mais dispersas no espaço de busca são priorizadas. Quando torna-se necessário eliminar indivíduos do repositório, opta-se por partículas pertencentes a regiões mais densas, consideradas redundantes. Os autores incluem também um fator de turbulência para que as partículas escapem de ótimos locais. Dois problemas com dois objetivos e um problema com três objetivos são avaliados e comparados com duas implementações para problemas multiobjetivo baseadas em algoritmos genéticos. Uma métrica para diversidade e outra para convergência são utilizadas e os autores concluem que o DMOPSO apresenta os melhores resultados. Porém, considerando que os resultados são altamente dependentes da função a ser otimizada, o trabalho falha ao não apresentar mais problemas com mais que dois objetivos. Além disso, a dimensão dos problemas poderia ser alterada, explorando a dificuldade dos mesmos e outras métricas poderiam ser consideradas.

Fdhila et al. [Fdhila et al., 2011] propõem um MOPSO distribuído para resolver um problema de seleção de características (do inglês, *feature selection problem*), tam-

bém conhecido como “*subset selection*” que geralmente é utilizado no aprendizado de máquina. O problema abordado é composto por dois objetivos de minimização conflitantes entre si e os resultados obtidos pelo MOPSO proposto são comparados com o NSGA II [Deb et al., 2002a]. O algoritmo divide a fronteira de Pareto em um conjunto com vinte fronteiras independentes. Cada subdivisão está associada a um enxame que executa o MOPSO independentemente e, a cada ciclo, atualiza um repositório de soluções não dominadas. A escolha do líder segue um padrão baseado em densidade de partículas. Os autores concluem, empiricamente, que os melhores resultados são obtidos pela proposta apresentada. Porém, a análise de resultados é pobre, são apresentados valores referentes a apenas 10 e 50 iterações e aparentemente, apenas os resultados de uma execução foram expostos. Nenhum teste estatístico é considerado e o que poderia ser um importante aspecto a ser explorado (os múltiplos enxames) é superficialmente tratado. Os múltiplos enxames otimizam regiões da fronteira de Pareto e nada é afirmado sobre a relação existente entre os enxames. Cada enxame atualiza apenas um repositório. Além disso, basicamente, um único problema com dois objetivos e três bases de dados é considerado. Métricas matemáticas para avaliar a diversidade ou convergência não são aplicadas e a avaliação baseada em tempo de execução não contempla um conjunto de rodadas para que média ou desvio padrão possam ser analisados.

Lee e Kim [Lee e Kim, 2011] abordam a resolução de problemas multiobjetivo aplicando uma classificação baseada na preferência do usuário. Para cada objetivo, o usuário determina um grau de preferência ou consideração que é representado por medidas difusas (do inglês, *fuzzy measures*). Os autores enfatizam que em problemas multiobjetivo, o conjunto de soluções potenciais é muito grande, pois existem muitas soluções não dominadas. Na prática, o usuário está interessado em uma solução desse conjunto. Desse modo, é proposto o algoritmo MOPSO-PS (do inglês, *Multi-Objective Particle Swarm Optimization with Preference-based Sorting*). Para cada partícula, é definida uma medida de avaliação, baseada em métodos para calcular integrais *fuzzy*. Assim, é determinado o *pbest* da partícula. O melhor valor (*gbest*) é

obtido do repositório de soluções não dominadas em que é aplicada a função de agrupamento (*crowding distance*) para estabelecer uma classificação com as soluções mais dispersas. Um indivíduo pertencente a 1/4 das soluções mais dispersas é escolhido para ser o líder (*gbest*). Os autores avaliam um problema multiobjetivo e comparam os resultados com outros três algoritmos. O problema em questão considerou sete objetivos, sendo três objetivos preferenciais. Os resultados apresentados apontam soluções que priorizam esses três objetivos. O trabalho apresenta um enfoque prático adequado quando sabe-se qual objetivo é mais importante que outro. Porém, com vários objetivos, essa a asserção não é trivial. Outro aspecto negativo da proposta é o custo computacional envolvido para avaliar o *pbest* a cada iteração. Finalmente, o problema considerou vetores solução com um tamanho de apenas 11 dimensões e duas métricas foram utilizadas (*hypervolume* e *spread*).

Em [de Carvalho e Pozo, 2011a], Carvalho e Pozo observam que a convergência e a diversidade das soluções obtidas pelos MOPSOs são determinadas pela cooperação entre as partículas, nesse caso, fortemente influenciada pela escolha de líderes. Dependendo dessa escolha, os resultados podem induzir a resultados que priorizem a convergência ou a diversidade. A escolha dos líderes é realizada a partir do repositório das soluções não dominadas e os autores incorporam a técnica CDAS (do inglês, *Controlling Dominance Area of Solutions*), proposta por [Sato et al., 2007] para modificar a relação de dominância entre as soluções do repositório. Essa técnica controla o nível de contração e de expansão da área de dominância das soluções através de uma equação trigonométrica. Assim, com essa contração ou expansão, soluções originalmente não dominadas tornam-se dominadas por outras e vice-versa. Essa modificação é implementada na atualização do repositório de dois algoritmos MOPSO e três funções de *benchmarks* são executadas. Diversas modificações no controle de dominância (configurado por um parâmetro) são realizadas e os resultados são analisados utilizando várias métricas para aferir a convergência e a diversidade dos resultados. Os autores concluem que, na maioria das vezes, alterando o controle de dominância de modo a gerar soluções que anteriormente seriam não dominadas em soluções do-

minadas melhora os resultados nos dois algoritmos. Os autores destacam também que, sem a aplicação do CDAS, os algoritmos originais têm sua habilidade de busca deteriorada à medida que o número de objetivos aumenta.

Motivados pelos resultados promissores obtidos em [de Carvalho e Pozo, 2011a], os autores ampliam [de Carvalho e Pozo, 2011b] o referido estudo aplicando uma técnica chamada de AR [Bentley e Wakefield, 1998] (do inglês, *Average Ranking*). Essa técnica determina uma ordem de preferência no conjunto de soluções não dominadas. O AR calcula independentemente uma classificação para cada valor objetivo. Após o cálculo de cada *ranking*, o AR é a soma todos esses *rankings*. Assim, o algoritmo escolhe as melhores soluções (líderes) através dessa ordem de preferência e não pela relação de dominância tradicional: uma solução  $x$  domina uma solução  $y$ , denotado por  $x \prec AR y$ , se e somente se  $AR(x) < AR(y)$ . Os autores realizam análises empíricas para avaliar o desempenho de um algoritmo MOPSO usando duas relações de preferência: CDAS e AR. O algoritmo utilizado é o SMPSO [Nebro et al., 2009] e assim, duas extensões desse algoritmo são implementadas: CDAS-SMPSO e AR-SMPSO. Tais algoritmos são aplicados em dois *benchmarks* de problemas com muitos objetivos e três indicadores de qualidade foram utilizados. Os resultados obtidos pelas melhores configurações dos três algoritmos foram confrontados e os autores concluem que o CDAS-SMPSO apresenta o melhor desempenho. Porém, os autores destacam que o AR-SMPSO apresentou melhor desempenho que o SMPSO original e, ao contrário do CDAS-SMPSO, ele não é dependente da configuração de parâmetros adicionais.

#### 4.4.1 MOPSOs baseados em decomposição

O bom desempenho apresentado pelo MOEA/D para resolver MOPs motivou o desenvolvimento de adaptações de seu algoritmo para operadores não genéticos, inclusive para MOPSOs. O primeiro MOPSO baseado em decomposição, chamado de MOPSO/D foi apresentado por Peng e Zhang [Peng e Zhang, 2008] em 2008. Baseado no *framework* do MOEA/D, o MOPSO/D substitui os operadores genéticos (cruzamento e mutação) por equações de movimentação de partículas do PSO tradicional.

Um fator de turbulência é utilizado de maneira análoga à mutação do MOEA/D e a estratégia de arquivamento das soluções não dominadas é baseada na  $\epsilon$ -dominância proposta por [Laumanns et al., 2002]. Para a decomposição dos MOPs em problemas escalares, os autores utilizam o método de *Tchebycheff*. A atualização da velocidade das partículas no MOPSO/D é baseada na equação da velocidade com peso de inércia  $\omega$  proposta por [Shi e Eberhart, 1998]:

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + C1 \cdot rand() \cdot (\vec{pbest}_i(t) - \vec{x}_i(t)) + C2 \cdot rand() \cdot (\vec{gbest}_i(t) - \vec{x}_i(t)) \quad (4.1)$$

Neste algoritmo, o *pbest* é simplesmente a melhor posição atual da partícula e a escolha do *gbest* utiliza a noção de vizinhança estabelecida pelo MOEA/D. Ou seja, considerando a menor distância entre os vetores de pesos e a partícula a ser atualizada,  $T$  vizinhos são definidos. Calcula-se a aptidão desses  $T$  vizinhos e a solução com melhor desempenho é escolhida como *gbest*. Os autores executam uma série de problemas bi-objetivos e comparam os resultados do MOPSO/D com os obtidos por uma variação do NSGA-II [Deb et al., 2002a]. Os autores concluem que para a maioria das funções, o MOPSO/D apresenta melhores resultados. Porém apenas problemas com dois objetivos foram testados e apenas uma métrica de desempenho foi utilizada (no caso, o IGD). Ademais, muitas questões relacionadas aos MOPSOs disponíveis na literatura não são abordadas.

Outra proposta de MOPSO baseada em decomposição foi publicada por Moubayed *et al.* [Moubayed et al., 2010]. O algoritmo, chamado de SDMOPSO também é baseado no MOEA/D e utiliza um arquivo externo baseado na  $\epsilon$ -dominância proposta por [Laumanns et al., 2002] para manter a diversidade no enxame. Os autores analisam o trabalho de Peng e Zhang [Peng e Zhang, 2008] e apontam algumas limitações, destacando o mecanismo simples adotado pelo MOPSO/D para selecionar a melhor solução individual para cada partícula. Também ponderam que o MOPSO/D não incorpora o conceito de dominância dentro do processo de otimização, o qual pode

levar à convergência prematura. Considerando tais limitações, a maior contribuição do SDMOPSO é o uso do conceito de dominância de Pareto para conceber um procedimento mais cuidadoso de atualização do *pbest* de cada partícula. Para definir a nova velocidade de uma partícula, o algoritmo utiliza a posição atual da partícula e seleciona aleatoriamente um vizinho para ser usado como *pbest*. Após a atualização da posição da partícula, o valor do *pbest* será substituído pela nova posição apenas se a nova posição dominar o *pbest* atual. Em relação aos demais procedimentos, os autores utilizam os mesmos mecanismos presentes no MOEA/D, inclusive a escalização baseada no método de *Tchebycheff*. O algoritmo SDMOPSO é testado em problemas com dois e três objetivos e os resultados são comparados com o MOEA/D e com o OMOPSO (um MOPSO baseado em  $\epsilon$ -dominance proposto por Reyes-Sierra e Coello [Sierra e Coello, 2005]). Os autores concluem que, para dois objetivos, os três algoritmos apresentam desempenho similar. Para três objetivos, o MOEA/D é superado pelos outros dois algoritmos. Embora o MOPSO/D, criticado pelos autores, não tenha sido considerado nos testes, deduz-se que a implementação de mecanismos mais elaborados para ditar a movimentação das partículas pode exercer um efeito positivo na otimização. A escalabilidade do algoritmo para mais que três objetivos não é considerada e o potencial do MOPSO não é completamente explorado. Os próprios autores que propuseram o SDMOPSO desenvolveram um novo algoritmo, chamado D<sup>2</sup>MOPSO [Al Moubayed et al., 2012], com o intuito de melhorar a diversidade e convergência dos pontos, utilizando técnicas de arquivamento baseadas em *Crowding Distance*.

Após analisar os MOPSOs baseados em decomposição e observar que a combinação entre decomposição e PSO não foi devidamente explorada, Martínez e Coello propuseram o *decomposition-Based MOPSO* (dMOPSO) [Zapotecas Martínez e Coello Coello, 2011]. O dMOPSO tem como principais características:

- não empregar fatores de turbulência nem utilizar o conceito de dominância de Pareto;
- selecionar líderes com base na natureza do próprio método de decomposição;

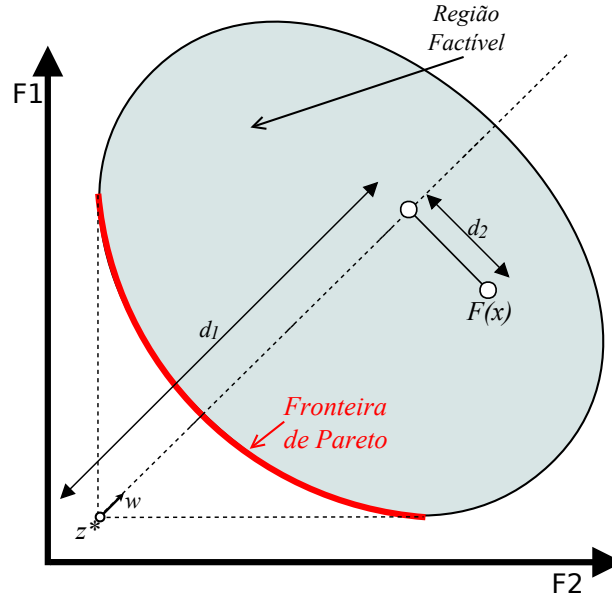


Figura 4.1: Ilustração do método PBI - *Penalty Boundary Intersection* [Zapotecas Martínez e Coello Coello, 2011].

- reiniciar as partículas baseada em sua idade (partículas que não evoluem, envelhecem);
- não utilizar arquivo externo para manter soluções não dominadas (um *gbest* é associado para cada vetor de peso; assim, existe um conjunto de *gbest*'s).

O dMOPSO utiliza como função de escalarização uma variação do método *Boundary Intersection*, chamado de *Penalty Boundary Intersection* - PBI. O PBI usa um vetor de pesos  $w$  e um valor de penalidade  $\theta$  para minimizar as distâncias da solução  $F(\vec{x})$  tanto para o *vetor ideal*  $d_1$  quanto para o erro de direção associado ao vetor de peso  $d_2$ . A Figura 4.1 ilustra esta abordagem. Assim, considerando método PBI, os autores definem um problema de otimização como:

$$\text{minimizar: } g(\vec{x}|\vec{w}, z^*) = d_1 + \theta d_2 \quad (4.2)$$

onde  $d_1 = \frac{|(F(\vec{x}) - z^*)^T \vec{w}|}{\|\vec{w}\|}$  e  $d_2 = \|(F(\vec{x}) - z^*) - d_1 \frac{\vec{w}}{\|\vec{w}\|}\|$ ,  $\vec{x} \in \mathfrak{R}^N$  e  $z^* = \min f_i(\vec{x}) | \vec{x} \in \Omega$ .

Assim, uma boa representação da fronteira de Pareto pode ser obtida pela definição de um conjunto de vetores de pesos bem distribuídos. A seguir, o Algoritmo 6 demonstra o pseudo-código do dMOPSO.



---

**Algoritmo 6:** Pseudo-código do dMOPSO [Zapotecas Martínez e Coello Coello, 2011].

---

**Entrada:** Problema de Otimização.

**Saída:** Conjunto  $G_{best}$ .

- 1: **ETAPA 1: Inicialização:**
  - 2:  $t \leftarrow 0$ ; //número de ciclos
  - 3: gere um conjunto de  $N$  vetores de peso  $W \leftarrow \{w_1, \dots, w_N\}$ ;
  - 4: gere aleatoriamente um enxame  $P^t \leftarrow \{x_1, \dots, x_N\}$  de  $N$  partículas;
  - 5: inicialize a velocidade  $v_i$  e a idade  $a_i$  de cada partícula  $x_i$  com zero, i.e.,  
 $v_i^t \leftarrow a_i^t \leftarrow 0$ , (para  $i=1, \dots, N$ );
  - 6: defina o  $pbest$ :  $x_{pb,i} \leftarrow x_i$ , (para  $i=1, \dots, N$ );
  - 7: defina o conjunto de  $gbest$ 's:  $G_{best} \leftarrow P^t$ .
  - 8: **ETAPA 2: misture o conjunto de  $gbest$ 's.**
  - 9: **ETAPA 3: o ciclo:** // **atualize**  $P^{t+1} \leftarrow \{x_1^{t+1}, \dots, x_N^{t+1}\}$
  - 10: **para**  $i = 1, \dots, N$  **faça**
  - 11: **se** ( $a_i^t < T_a$ ) **então**
  - 12: **atualize a partícula:** atualize a velocidade  $v_i^{t+1}$  (usando como  $gbest$  a solução  $x_{gb,i} \in G_{best}$ ) e atualize a posição  $x_i^{t+1}$  de acordo com as Equações 3.3 e 3.1, respectivamente.
  - 13: **senão restabeleça a partícula:** reinicie a memória da partícula, i.e.,  
 $v_i^{t+1} = a_i^{t+1} = 0$  e reinicie a posição de  $x_i^{t+1}$  de acordo com a Equação 4.3.
  - 14: **repare os limites** da partícula  $x_i^{t+1}$  e atualize a velocidade  $v_i^{t+1}$  de acordo com a equação 4.4;
  - 15: **avale a partícula e atualize  $z^*$ :** obtenha  $F(x_i^{t+1})$ . **se**  $f_j(x_i^{t+1}) < z_j^*$  **então**  
 $z_j^* \leftarrow f_j(x_i^{t+1})$ , (para  $i=1, \dots, N$ );
  - 16: **atualize o  $pbest$ :** **se**  $g(x_i^{t+1}|w_i, z^*) \leq g(x_{pb,i}^{t+1}|w_i, z^*)$  **então**  $x_{pb,i} \leftarrow x_i^{t+1}$  e  $a_i^{t+1} \leftarrow 0$   
**senão**  $a_i^{t+1} \leftarrow a_i^t + 1$ .
  - 17: **fim-para**
  - 18: **ETAPA 4: atualize o  $gbest$ :** Obtenha  $G_{best}$  de  $S \leftarrow G_{best} \cup P^{t+1}$ , i.e.,  
**atualizeGlobalBest(W, P).**
  - 19: **ETAPA 5: critério de parada:** **se** ( $t < N_{ger}$ ) **então**  $t \leftarrow t + 1$  e retorne ao **PASSO 2**. Caso contrário, pare e **retorne** o conjunto  $g_{best}$ .
- 

Inicialmente, um conjunto de  $N$  partículas  $P \leftarrow \{x_1, \dots, x_N\}$  é inicializado aleatoriamente.  $Pbest$  é definido como  $x_{pb,i} = x_i$ . Cada partícula possui uma velocidade  $v_i$  e uma idade  $a_i$ , ambas inicializadas com zero. No decorrer dos ciclos de “vôo”, cada partícula tenta minimizar um dos subproblemas definidos por cada vetor de pesos  $w_i$ . Assim, cada partícula voa em direção à melhor posição, de modo a minimizar um subproblema mono-objetivo  $g(x_i|w_i, z^*)$ . O  $pbest$   $x_{pb,i}$  da  $i_{esima}$  partícula representa a melhor posição da partícula para o  $i_{esimo}$  subproblema.

A cada ciclo, as melhores posições de cada partícula são usadas para encon-

---

**Algoritmo 7:**  $G_{best} \leftarrow \text{atualizeGlobalBest}(W, P)$  - Etapa 4 do dMOPSO.

---

```

1:  $T \leftarrow P, G \leftarrow \emptyset$ ;
2: para todo  $w_i \in W$  faça
3:    $G \leftarrow G \cup \{x_j | \min_{x_j \in T} g(x_j | w_i, z^*)\}$ ;
4:    $T \leftarrow T \setminus \{x_j\}$ ;
5: fim-para
6: retorne  $G$ 

```

---

trar as melhores soluções para cada subproblema. Portanto, o conjunto de melhores soluções globais  $G_{best}$  é definido naturalmente. Este conjunto conterá as soluções que minimizam cada subproblema e é atualizado em cada ciclo de acordo com o Algoritmo 7. Uma vez que o conjunto de  $gbest$ 's esteja definido, a velocidade e a posição de cada partícula são atualizadas de acordo com as equações tradicionais do PSO, definidas em 3.3 e 3.1, respectivamente. O dMOPSO assume que todas as soluções de  $G_{best}$  são igualmente boas, portanto, a velocidade de cada partícula é calculada usando como seu  $gbest$  uma solução escolhida aleatoriamente de  $G_{best}$ . A idade da partícula promove a diversidade ao longo da otimização e indica quando uma partícula não fornece boas informações em seu processo de “vôo”. Se uma partícula não melhora seu  $pbest$ , então ela envelhece ( $a_i$  é incrementada em 1). E, caso a partícula ultrapasse um limite pré-definido (no dMOPSO,  $T_a = 2$ ), os valores de  $v_i$ ,  $a_i$  e  $x_{pb,i}$  são reiniciados e a nova posição da partícula é calculada de acordo com a Equação 4.3:

$$x_i^{t+1}(j) = N\left(\frac{x_{gb,i}(j) - x_{pb,i}(j)}{2}, |x_{gb,i}(j) - x_{pb,i}(j)|\right), \quad (4.3)$$

onde  $N(m, \theta)$  representa um número aleatório usando uma distribuição normal com média  $m$  e desvio padrão  $\theta$ . Em PSO, existe a possibilidade da partícula ultrapassar os limites estabelecidos pelo problema. Para tratar esse problema, o dMOPSO utiliza a Equação 4.4 para reestabelecer a posição  $x_i^{t+1}$  e a velocidade  $v_i^{t+1}$  em função do  $j_{esimo}$  limite:

$$\begin{aligned}
 x_i^{t+1}(j) &= \begin{cases} x_{lb}(j), & \text{se } x_i^{t+1}(j) < x_{lb,i}(j) \\ x_{ub}(j), & \text{se } x_i^{t+1}(j) > x_{ub,i}(j) \end{cases} \\
 v_i^{t+1}(j) &= -\Upsilon x_i^{t+1}(j),
 \end{aligned} \tag{4.4}$$

onde  $x_{lb,i}(j)$  e  $x_{ub,i}(j)$  são, respectivamente, os limites inferior e superior do  $j_{esimo}$  componente dos valores permitidos das variáveis de decisão; e  $\Upsilon = 1$ .

Os autores comparam o desempenho do dMOPSO entre o MOEA/D [Zhang e Li, 2007] e o SDMOPSO [Moubayed et al., 2010] para nove problemas. A maioria dos problemas são testados com dois objetivos e em três problemas, três objetivos são avaliados. Os autores utilizam *hypervolume* [Zitzler, 1999], *spacing* [Schott, 1995] e *Coverage of Two Sets* [Zitzler et al., 2000] como indicadores de desempenho para avaliar os algoritmos. Os autores defendem que a maioria dos resultados indicam que o dMOPSO foi superior aos outros dois algoritmos, excetuando poucas e pontuais situações. Por exemplo, em três funções o MOEA/D gerou mais soluções não dominadas que o dMOPSO, porém tais soluções ficaram restritas a uma única região da fronteira de Pareto real.

O dMOPSO mostra-se como um algoritmo extremamente competitivo, todavia sua escalabilidade não foi devidamente avaliada no contexto de problemas com vários objetivos. Entendemos também que o uso de subpopulações é uma estratégia que merece ser investigada, assim como as relações associadas à conectividade entre os enxames e a relação de compartilhamento de experiências entre as mesmas. Assim, propomos um MOPSO baseado em decomposição que estabelece uma variação do dMOPSO para resolver problemas com vários objetivos em um ambiente distribuído. Esse novo MOPSO distribuído baseado em decomposição é tratado na Seção 5.2 do próximo Capítulo.

## CAPÍTULO 5

### ESTRATÉGIAS DISTRIBUÍDAS PARA A RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO UTILIZANDO O ALGORITMO PSO

Neste capítulo propomos abordagens cooperativas baseadas no algoritmo PSO para a resolução de problemas de otimização. O capítulo é dividido em duas seções principais, que abordam duas classes de problemas de otimização distintas. A primeira seção descreve uma abordagem distribuída para a resolução de funções de otimização baseadas em *benchmarks* mono-objetivos. Tal abordagem envolve a resolução de problemas de otimização com um objetivo e fundamenta a contribuição principal desse trabalho que é a resolução de problemas com vários objetivos considerando o uso de múltiplos enxames na otimização por enxame de partículas.

A segunda seção deste capítulo trata das funções de otimização de *benchmarks* com vários objetivos. Apresentamos duas propostas para resolver problemas com vários objetivos. A primeira compreende um PSO distribuído para resolver MaOPs baseado em conceitos de dominância de Pareto tradicionais. A segunda proposta é uma estratégia distribuída para a resolução de problemas com vários objetivos baseado em decomposição.

#### 5.1 Uma abordagem assíncrona para a resolução de problemas mono-objetivo utilizando o algoritmo PSO

Nessa seção descrevemos uma técnica para a execução de múltiplos enxames em várias topologias usando um método de comunicação assíncrono. As partículas são organizadas em enxames (subpopulações) e, em cada enxame, todas estão completamente conectadas. O propósito da adoção desta estratégia é permitir que cada população possa executar o processo de otimização independentemente, em uma tentativa

de acelerar o processo de otimização e também prevenir a convergência para soluções subótimas [Løvberg et al., 2001]. Pela adoção de múltiplos processadores executando os enxames independentemente, o processo de otimização pode ser executado em paralelo, que pode representar uma estratégia adequada para otimizar funções com alta dimensionalidade. Usando a taxonomia introduzida em [El-Abd e Kamel, 2008] e apresentada no Capítulo 3, propomos um algoritmo distribuído baseado em PSO considerando, respectivamente, as seguintes respostas para cada questão anteriormente apresentada:

1. **Qual informação trocar?** a informação trocada é a melhor partícula de cada enxame;
2. **Quando trocar a informação?** a estratégia de comunicação adotada é assíncrona. Isso é, o compartilhamento não ocorre em intervalos fixos. A melhor partícula de cada enxame é enviada apenas quando ocorre uma melhora no *gbest* no enxame;
3. **Como compartilhar a informação?** a informação é compartilhada considerando uma gama de possibilidades, dependente da topologia de vizinhança adotada. As topologias utilizadas são *broadcast*, anel unidirecional, anel bidirecional, *gossip* (definido a seguir) para um vizinho, *gossip* para  $\log(N)$  vizinhos (todos os logaritmos deste trabalho são base 2), o hipercubo e, finalmente, uma topologia dinâmica;
4. **O que fazer com a informação trocada?** o *gbest* recebido de outro enxame é analisado com o *gbest* do próprio enxame. Caso, a partícula recebida seja melhor, então este passa a ser o *gbest* corrente.

Consideramos como componente social o *gbest* que é compartilhado entre enxames de maneira assíncrona. Para cada função de otimização foi estabelecido um valor de referência. O procedimento de otimização (PSO padrão) é executado isoladamente em um processador. Em seguida, o processo é repetido em um ambiente simulado

utilizando 16 enxames. Ao contrário do modelo centralizado, em que o melhor resultado está constantemente disponível a todas as partículas, no modelo cooperativo é necessário um mecanismo que possibilite compartilhar o componente social da PSO entre os enxames. A população de partículas é distribuída em vários enxames. Cada enxame executa o processo de otimização independentemente. Um *gbest* é computado por enxame e este *gbest* exerce influência apenas na posição das partículas do próprio enxame. Essa informação é trocada entre diferentes enxames. Para todas as estratégias de comunicação/topologias adotadas, o procedimento de compartilhamento de informação adotado é o seguinte:

- quando um enxame melhora seu *gbest*, ele escolhe um ou mais vizinhos (dependendo do método de compartilhamento) e envia o seu *gbest* e seu *fitness*;
- o enxame que recebe tais informações compara com seus próprios *gbest/fitness*;
- se o *gbest* recebido for o melhor, o enxame assume este como seu melhor global; caso contrário, a informação é ignorada.

Esse mecanismo garante que os enxames compartilhem entre si o componente social do procedimento de otimização. Esse processo acaba quando o resultado atingir um erro menor que um determinado valor ou quando um limite de iterações é atingido. Assim, é possível investigar a relação entre múltiplos enxames e a topologia de vizinhança adotada na influência dos resultados em processos de otimização utilizando o algoritmo PSO. Como anteriormente mencionado, a definição da melhor solução da vizinhança depende da topologia de vizinhança implementada. Nesse trabalho, investigamos o desempenho de múltiplos enxames distribuídos em diferentes topologias e comparamos com a PSO padrão *fully connected*. As seguintes estratégias de comunicação/topologias foram adotadas para que os enxames compartilhem o componente social (*gbest*):

1. **broadcast**: a informação é enviada de uma origem para todos os destinos. O componente social é enviado de um enxame para todos os outros enxames; estratégia baseada em uma topologia de grafo completo;

2. **anel unidirecional**: estratégia baseada em uma topologia em anel, o componente social é enviado de um enxame  $i$  para seu sucessor no anel;
3. **anel bidirecional**: estratégia também baseada em uma topologia em anel, o componente social é enviado para o predecessor e sucessor do enxame  $i$ ;
4. ***gossip* para 1 vizinho**: protocolos *gossip* (ou protocolos epidêmicos) são amplamente usados como estratégias para disseminação de informações em redes de computadores [Van Renesse et al., 1998]. A nova informação é enviada a destinos escolhidos aleatoriamente e o número de destinos é chamado de *fan-out*. Esta estratégia é baseada em uma topologia de grafo completo, e o *fan-out* é igual a 1, isto é, um vizinho é escolhido aleatoriamente para receber o componente social;
5. ***gossip* para log (N) vizinhos**: similar ao método anterior, baseado em uma topologia de grafo completo, neste caso o *fan-out* é  $\log(N)$ , isto é,  $\log(N)$  enxames são escolhidos aleatoriamente para receber o componente social. Nos exemplos, como  $N=16$ ,  $\log(N) = 4$ ;
6. **hipercubo**: na topologia hipercubo, cada enxame é conectado a  $\log(N)$  outros enxames, de modo que cada enxame é adjacente à outro quando seus identificadores diferem apenas um *bit*. O componente social é enviado de um determinado enxame para seus  $\log(N)$  vizinhos.
7. **dinâmica**: a topologia dinâmica utilizada nesse trabalho foi introduzida por Oca et al. [Montes de Oca et al., 2009] e é uma topologia que varia ao longo do tempo. Inicialmente, todos os enxames estão conectados entre si e a mudança da topologia é escalonada em um número de iterações pré determinado. Considerando  $n$  como o número de partículas, uma topologia, inicialmente, de grafo completo com  $n(n-1)/2$  arestas se transformará em uma topologia em anel com  $n$  arestas. Assim, a conectividade é reduzida ao longo do tempo. Esse conceito foi adaptado a este trabalho, em que a conectividade é tratada entre

**Algoritmo 8:** Pseudo-código do algoritmo da topologia dinâmica.

---

```

1: // Inicialização
2:   $t \leftarrow 0$ ;  $esteps \leftarrow 0$ ;
3:   $k \leftarrow$  iterações necessárias para alterar a topologia;
4: // Procedimento incluído no laço do PSO
5: // Atualização da topologia
6: se  $(t > 0) \wedge (t \leq k) \wedge (t \bmod (k/(n-3)) = 0)$  então
7:   para  $i = 1$  até  $n - (2 + esteps)$  faça
8:    //  $n - (2 + esteps)$  garante o padrão de regressão aritmética
9:    Escolha aleatoriamente dois enxames  $i$  e  $j$  não adjacentes
10:   Elimine a aresta entre  $i$  e  $j$ 
11:   fim para
12:   $esteps \leftarrow esteps + 1$ 
13: fim se
14:  $t \leftarrow t + 1$ 

```

---

subpopulações. O procedimento de remoção de arestas é incluído no processo de otimização e seu pseudocódigo é ilustrado no Algoritmo 8, descrito a seguir.

Considerando o Algoritmo 8,  $t$  representa o tempo atual (iteração) e inicia em 0. A variável  $esteps$  garante o padrão de regressão aritmético e também inicia em 0.  $k$  representa o intervalo necessário (medido em iterações) para que arestas sejam removidas. A seguir, no laço do algoritmo PSO um teste lógico é realizado, onde:

- $t > 0$  garante que uma topologia de grafo completo é inicialmente utilizada;
- $t \leq k$  garante que o processo de atualização da topologia não é executado após a iteração  $k$ ;
- $t \bmod (k/(n-3)) = 0$  garante o correto escalonamento do procedimento de atualização da topologia.

A Figura 5.1 ilustra quatro das topologias que definem as relações entre os enxames: anel unidirecional, anel bidirecional, grafo completo e hipercubo. Os vértices representam os enxames e as arestas representam as relações entre os enxames. Tanto quanto sabemos e acreditamos, as topologias em hipercubo e dinâmica nunca foram anteriormente usadas como topologias para conectar múltiplas subpopulações na otimização por enxame de partículas.



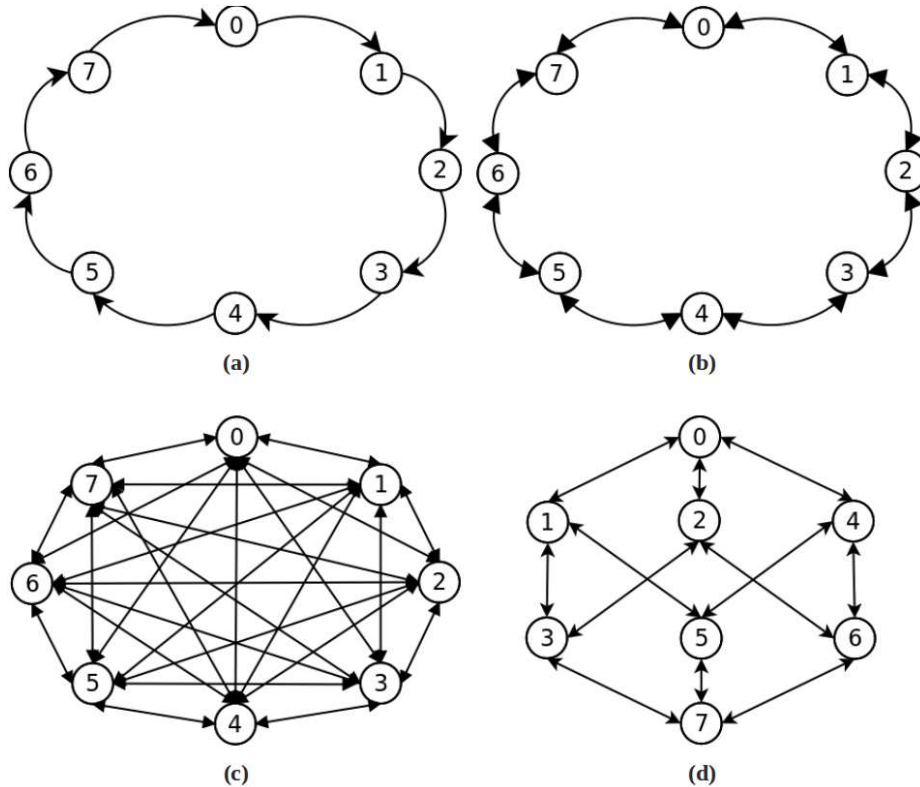


Figura 5.1: Topologias de enxames: **(a)** anel unidirecional; **(b)** anel bidirecional; **(c)** grafo completo; **(d)** hipercubo.

Na Figura 5.1, são ilustrados oito enxames independentes, representados por círculos. É importante frisar que três diferentes abordagens foram usadas para trocar o componente social na topologia com grafo completo: *broadcast*, *gossip* para um vizinho e *gossip* para  $\log(N)$  vizinhos.

A seguir, apresentamos métodos baseados em PSO, nos quais os enxames são distribuídos e cooperam para resolver problemas com mais de um objetivo.

## 5.2 Algoritmos baseados em múltiplos enxames distribuídos para resolver problemas com vários objetivos

Nesta seção apresentamos duas propostas para resolução de problemas de otimização utilizando o PSO com múltiplos enxames. A primeira proposta é um PSO distribuído para resolver MaOPs baseado em conceitos tradicionais das relações de dominância e não dominância entre soluções. A segunda proposta é uma estratégia

distribuída para a resolução de problemas com vários objetivos baseado na decomposição de problemas multiobjetivo em subproblemas mono-objetivo.

### 5.2.1 Estratégia distribuída baseada em conceitos de dominância para a resolução de MaOPs

Aqui propomos um MOPSO distribuído que emprega múltiplos enxames para resolver problemas com vários objetivos. A população de partículas é distribuída em vários enxames e, em cada enxame, todas as partículas são conectadas entre si. Cada enxame evolui independentemente e mantém seu próprio repositório de soluções não dominadas. A interação entre os enxames ocorre pelo compartilhamento de conjuntos de soluções não dominadas e consideramos duas estratégias de interconexão entre enxames para estabelecer a migração de tais conjuntos.

Para desenvolver um MSPSO para resolver problemas com vários objetivos, nos baseamos na taxonomia apresentada por El-Abd e Kamel [El-Abd e Kamel, 2008] e a adaptamos ao contexto de múltiplos objetivos. Respondemos às questões elencadas na referida taxonomia da seguinte forma:

- a) **A informação trocada corresponde a um “*buffer*”:** cada enxame possui um conjunto de soluções (partículas) não dominadas que é mantido em um *buffer*. A cada passo, novas soluções não dominadas do enxame são adicionadas ao *buffer*. A seguir, este *buffer* é enviado a enxames vizinhos e esvaziado.
- b) **A estratégia de comunicação é assíncrona:** o *buffer* de cada enxame é enviado apenas se seu conteúdo é melhorado. Assim, ao fim de cada iteração, o *buffer* é enviado somente se novas soluções não dominadas são encontradas e adicionadas ao *buffer*.
- c) **Duas estratégias para compartilhar a informação são propostas e avaliadas:** *broadcast*: o componente social (mantido no *buffer*) é enviado de um enxame para todos os outros enxames através de uma topologia de grafo completo; *gossip*: também chamado de protocolo epidêmico, é muito utilizado para

a disseminação de informações em redes de computadores [Van Renesse et al., 1998]. O *buffer* é enviado para destinos escolhidos aleatoriamente. O número de tais destinos é chamado de *fanout*. Nesta estratégia consideramos uma topologia de grafo completo com *fanout* igual a 1.

- d) O repositório é atualizado:** o *buffer* recebido de outro enxame é avaliado levando em conta o conjunto de soluções não dominadas mantidas no repositório local. Se as partículas recebidas representam soluções não dominadas, então o repositório é atualizado.

Nossa concepção de MOPSO utiliza o método *sigma* [Mostaghim e Teich, 2003] para selecionar líderes e o CD (*Crowding Distance*) [Raquel e Naval, 2005] para gerenciamento do repositório de soluções não dominadas. Esta abordagem é apresentada no pseudo-código do Algoritmo 9.

---

**Algoritmo 9:** MOPSO Usando o valor  $\sigma$  com *Crowding Distance*.

---

```

para cada partícula  $i$  faça
  inicialize  $\vec{x}_i, \vec{v}_i, NSol = 0$  (total de soluções em A);
fim-para;
enquanto número máximo de iterações não atingido faça
  para cada partícula  $i$  faça
    calcule o valor de aptidão atual  $f(\vec{x}_i)$ ;
    se  $f(\vec{x}_i)$  domina  $f(pbest_i)$  então
       $pbest_i \leftarrow \vec{x}_i$ ;
    fim-se;
  fim-para;
  para cada partícula  $i$  faça
    calcule o valor  $\sigma$  os membros de A
    calcule o valor  $\sigma_i$  para a partícula  $i$ 
     $gbest \leftarrow$  membro de A com menor distância Euclidiana entre o valor  $\sigma_i$  e o valor  $\sigma$  do membro de A;
    calcule a velocidade  $\vec{v}_i$  de acordo com a equação 3.2;
    atualize a posição  $\vec{x}_i$  de acordo com a equação 3.1;
    calcule o valor de aptidão  $f(\vec{x}_i)$ ;
    se  $\vec{x}_i$  domina  $pbest_i$  então
       $pbest_i \leftarrow \vec{x}_i$ ;
    fim-se;
  atualize A de acordo com o Algoritmo 10
fim-para;
fim-enquanto.

```

---

## Compartilhamento de informações

Tanto uma versão centralizada, como versões distribuídas foram implementadas e avaliadas. Nas execuções distribuídas, consideramos a cooperação entre 8 enxames. Os múltiplos enxames empregaram as estratégias *broadcast* e *gossip* (Figura 5.2) para comunicar e, ao fim de cada iteração, o seguinte procedimento para compartilhar informações foi adotado por ambas as estratégias de comunicação:

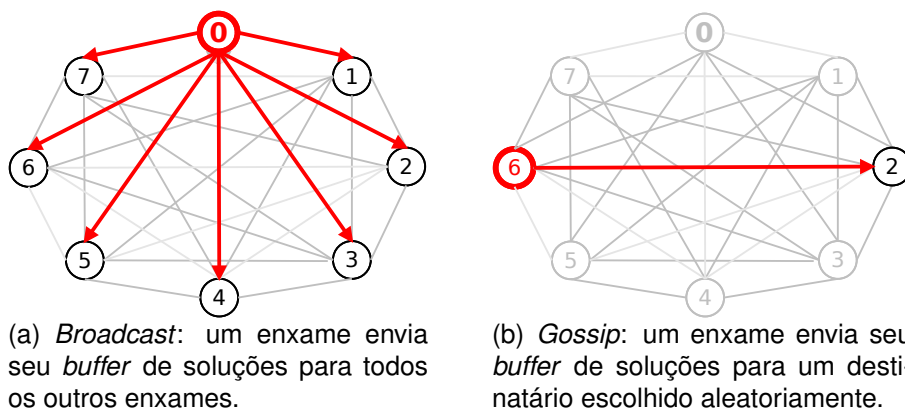


Figura 5.2: Compartilhamento de partículas: a seta indica a direção do compartilhamento de soluções entre os enxames remetente e destino(s).

- Quando um enxame melhora seus resultados, novas soluções são adicionadas a seu *buffer*. O tamanho do *buffer* foi limitado a 64 partículas. De modo a não exceder este valor, o CD é aplicado para eliminar soluções situadas em regiões mais densamente populosas;
- Logo em seguida, dependendo da topologia, se o *buffer* não está vazio, ocorre um evento de comunicação e um ou mais enxames vizinhos são escolhidos como destino e o *buffer* é enviado;
- Baseado na teoria de Pareto, o enxame destino compara o *buffer* recebido com suas próprias soluções mantidas em seu repositório local. Então, o repositório é atualizado considerando que cada solução no *buffer* pode estar em uma das seguintes situações:
  1. uma solução recebida **domina** alguma solução do repositório → a solução

recebida é armazenada no repositório e todas as soluções dominadas são eliminadas dele;

2. uma solução recebida é **dominada** por alguma solução do repositório → a solução recebida é ignorada;
3. **não existe relação de dominância** entre as soluções → se o repositório destino não está cheio (64 partículas), a solução recebida é adicionada. Se o repositório está cheio, então uma solução é substituída utilizando o cálculo da distância de agrupamento (CD). O Algoritmo 10 mostra como o repositório é atualizado; no pseudo-código, **A** corresponde ao arquivo externo (repositório).

Ao término de um número pré-definido de iterações, os algoritmos exibem o conteúdo dos conjuntos de soluções não dominadas.

---

**Algoritmo 10:** Atualização do repositório (A)

---

```

para cada partícula i faça
  se partícula i domina alguma solução em A então
    elimine a(s) solução(ões) de A
    adicione partícula i em A
  fim-se
  se nenhuma solução em A domina a partícula i então
    se A não está cheio então
      adicione partícula i em A
    senão
      substitua uma solução baseada no cálculo de seu CD
    fim-senão
  fim-se
fim-para.

```

---

A comunicação é desencadeada por um evento (no caso, a melhora dos resultados) e não por um número pré-determinado de iterações. Assim, esta estratégia de compartilhamento do componente social entre os enxames é classificada como *comunicação assíncrona*.

### 5.2.2 Estratégias distribuídas baseada em decomposição de problemas para a resolução de MaOPs

O MOEA/D é um *framework* recente baseado em GA, que decompõe um MOP em um número de subproblemas de otimização mono-objetivo [Zhang e Li, 2007]. O objetivo de cada subproblema é uma agregação ponderada de cada um dos múltiplos objetivos. É considerada uma técnica que atinge bons resultados e por conta de sua eficiência para resolver MOPs, tem sido adaptada a outras meta-heurísticas, como evolução diferencial (DE, do inglês Differential Evolution) [Venske et al., 2012] e ACO [Ke et al., 2013].

Em relação ao PSO, não tardou para que iniciativas baseadas em decomposição fossem propostas [Peng e Zhang, 2008, Moubayed et al., 2010, Zapotecas Martínez e Coello Coello, 2011, Al Moubayed et al., 2012]. Apesar das relevantes contribuições para a área, podemos destacar que tais MOEAs não investigam a escalabilidade dos algoritmos em termos de números de objetivos. Além disso, o uso de múltiplas populações não é ponderado. Considerando estas constatações, propomos estratégias baseadas em decomposição para resolver MaOPS utilizando múltiplos enxames.

Nossas estratégias foram baseadas no dMOPSO, proposto por Martínez e Coello [Zapotecas Martínez e Coello Coello, 2011]. Esta escolha justifica-se pelo desempenho obtido pelo algoritmo em relação aos algoritmos similares que representam o estado da arte na área. Os resultados publicados indicam que esta abordagem é competitiva e que superam o desempenho dos MOEAs comparados na maioria dos testes adotados. Aqui, não entraremos nos detalhes do funcionamento do dMOPSO, pois o mesmo foi descrito na Subseção 4.4.1 do Capítulo 4.

Em relação aos métodos que aqui propomos, novamente utilizamos a taxonomia apresentada por El-Abd e Kamel [El-Abd e Kamel, 2008] e concebemos duas estratégias distribuídas (baseadas em decomposição) para solucionar MaOPs. Tais métodos são similares entre si e diferem em relação à estratégia de comunicação adotada. Estabelecemos duas estratégias de comunicação: síncrona e assíncrona. Nos métodos

baseados nos conceitos de dominância de Pareto, cada enxame possui um “buffer” que funciona como um repositório de soluções não dominadas. O conteúdo deste “buffer” é enviado aos outros enxames apenas se contiver partículas. Após cada envio, este repositório é esvaziado. Isto estabelece um método assíncrono de comunicação, pois é condicionado a uma situação (“buffer” não vazio). Em nosso método baseado em decomposição, o conceito de “buffer” é inexistente e sempre existirá uma melhor partícula (*gbest*) associada a cada subproblema. Assim, a cada iteração, um enxame qualquer **sempre** terá informação para compartilhar, instituindo um arranjo de comunicação *síncrono*. De modo a estabelecer uma estratégia assíncrona para determinar **quando** compartilhar as informações, criamos um mecanismo que analisa se o *gbest* atual foi substituído. Assim, a cada iteração, um evento de comunicação ocorrerá apenas se a melhor solução associada a cada subproblema é nova. Concebemos ambos os métodos considerando:

- a) **A informação trocada corresponde a um conjunto de melhores soluções de cada enxame:** cada enxame possui um conjunto de melhores soluções (um *gbest* para cada subproblema). A cada passo, novas soluções são geradas a partir do “movimento” das partículas. Para cada subproblema, as partículas com os melhores valores de aptidão são definidas como o respectivo *gbest* do respectivo subproblema.
- b) **Duas estratégias de comunicação:** a) **síncrona:** a cada iteração o conjunto de melhores soluções (um *gbest* associado a cada subproblema) é compartilhado entre os outros enxames; b) **assíncrona:** o *gbest* de cada subproblema é enviado somente se esta partícula apresentar melhor aptidão que o *gbest* da iteração anterior.
- c) **Informação compartilhada por *broadcast*:** o componente social (estabelecido na forma de um conjunto de *gbests*) é enviado de um enxame para todos os outros enxames através de uma topologia de grafo completo;
- d) **O *gbest* de cada subproblema é atualizado:** o conjunto de soluções rece-

bido de outro enxame é avaliado com cada *gbest* associado a um determinado subproblema. Se as partículas recebidas representam soluções com melhores aptidões, então, para cada subproblema, o respectivo *gbest* é atualizado.

Baseado no dMOPSO [Zapotecas Martínez e Coello Coello, 2011], propomos uma estratégia distribuída para resolver MaOPS. Considerando o Algoritmo 7 do referido *framework*, definimos um método baseado em múltiplos enxames para resolver MaOPs conduzido da seguinte maneira:

- $S$  enxames são inicializados com uma população de  $N$  partículas em cada enxame;
- $W=S*N$  diferentes vetores de pesos são uniformemente definidos na forma  $W_i = (w_j, \dots, w_M)$ , para  $i=1, \dots, W$  e  $j=1, \dots, M$ . Onde  $M$  representa o número de objetivos do MOP, tal que  $w_j \geq 0$  e  $\sum_{j=1}^M w_j = 1$ ;
- os  $W$  vetores de pesos são ordenados crescentemente em função do  $M$ -ésimo objetivo. Após, para cada enxame são atribuídos, sequencialmente,  $N$  vetores de peso distintos. Um vetor de peso para cada subproblema  $\rightarrow$  um vetor de peso diferente para cada partícula;
- cada enxame executa o Algoritmo 7 independentemente, considerando sua população e seus vetores de peso por  $t$  iterações;
- cada partícula tenta minimizar um dos subproblemas, definido por seu vetor de peso  $\rightarrow$  uma partícula minimiza um subproblema mono-objetivo  $g(x_i|w_i, z^*)$  (utilizando a função de escalarização PBI, descrita na Subseção 4.4.1 do Capítulo 4);
- a cada iteração, em função da estratégia de comunicação (síncrona ou assíncrona), eventos de compartilhamento são desencadeados  $\rightarrow$  o conjunto de *gbest*'s de um enxame é enviado aos outros enxames;



- em cada exame destinatário, o conjunto de soluções recebido é unido ao conjunto de melhores soluções do próprio exame → para cada subproblema, mantém-se a solução (própria ou migrada) com melhor valor de aptidão computado;
- ao término de um número definido de  $t$  iterações, o conjunto de melhores soluções de cada exame é exibido.

No próximo capítulo validamos os métodos propostos através da execução de experimentos para avaliar o desempenho de nossas concepções para resolver problemas de otimização em múltiplos exames.

## CAPÍTULO 6

### AVALIAÇÃO EMPÍRICA DOS MÉTODOS PROPOSTOS

Neste capítulo validamos nossas estratégias distribuídas para a resolução de problemas de otimização utilizando o algoritmo PSO. Esta validação é realizada através da avaliação de execuções distribuídas para resolver funções de *benchmarks* que são comparadas às execuções centralizadas.

Este capítulo possui quatro seções principais. A primeira seção descreve a metodologia utilizada para validar nossas propostas. A segunda seção expõe a avaliação empírica de nosso método proposto para resolver problemas mono-objetivo. Finalmente, as terceira e quarta seções abordam os métodos apresentados para resolver problemas com vários objetivos. Estas duas últimas seções discorrem sobre estratégias baseadas em conceitos tradicionais de dominância de soluções e baseadas em decomposição de MOPs.

Os resultados experimentais da avaliação do desempenho da otimização com múltiplos exames distribuídos são descritos e discutidos.

#### 6.1 Metodologia para a validação dos métodos propostos

Neste trabalho propomos algoritmos baseados em PSO que utilizam múltiplos exames distribuídos para resolver problemas com um e vários objetivos. Nossa proposta foi executada em um ambiente simulado e funções de *benchmark* foram avaliadas.

Para medir a qualidade e avaliar o desempenho das técnicas propostas para resolver funções de otimização com um objetivo, analisamos os resultados obtidos em relação a cada ótimo global de cada função. Para problemas com mais de um objetivo, foram utilizados os indicadores de qualidade comuns à área e que são empregados para mensurar a convergência e diversidade das fronteiras obtidas. Nesta situação, testes de significância foram utilizados para apontar as diferenças observadas entre

os resultados obtidos através de testes estatísticos.

Nesta Seção expomos o ambiente simulado, os *benchmarks* utilizados e os testes estatísticos empregados.

### 6.1.1 Ambiente simulado

Nesta tese, consideramos a Otimização por Enxame de Partículas utilizando ambientes com múltiplos enxames. Tais ambientes foram simulados utilizando a ferramenta *SiMulation Programming Language* (SMPL), desenvolvida por MacDougall [MacDougall, 1987].

Segundo Sandhu [Sandhu, 1997], utilizar ambientes simulados apresenta as seguintes vantagens:

- permite o desenvolvimento de aplicações específicas e adaptadas às necessidades requeridas, resultando em economia de tempo para o desenvolvimento de um modelo;
- modelos podem ser facilmente modificados;
- tendência da programação requerer um menor número de linhas de código;
- fornecem mecanismos para verificar situações ilógicas ou impossíveis através de mensagens de erros específicas.

O SMPL é uma extensão funcional da linguagem de programação tradicional *C*. Essa ferramenta compõe uma linguagem de simulação fundamentada em três aspectos: *facilities*, *tokens* e *events*.

Uma *facility* representa algum recurso do sistema que está sendo modelado. Em nossa simulação, as duas *facilities* principais correspondem ao processo de otimização do algoritmo PSO e à atividade de comunicação entre os enxames.

Um *token* representa a entidade ativa do sistema. Basicamente, um *token* representará situações em que um enxame estará se comunicando com outros enxames ou evoluindo sua população de indivíduos.

Finalmente, os *events* correspondem às mudanças de estado no sistema. Em nossa simulação, eventos de comunicação são desencadeados pela melhora de indivíduos ou pela atualização no conteúdo do arquivo externo. Eventos de otimização são condicionados ao tempo decorrido na simulação, que estabelece o processo iterativo, propriamente dito.

### 6.1.2 Os *benchmarks* adotados

Neste trabalho, propomos a resolução de problemas de otimização considerando funções de *benchmark* relevantes. São utilizados problemas mono-objetivo e problemas multiobjetivo.

Para avaliar nossa proposta distribuída para resolver problemas com um objetivo, utilizamos um conjunto de seis funções de *benchmark* que foram consideradas em um espaço de busca multidimensional. As seguintes funções, amplamente referenciadas na literatura [Tang et al., 2007] foram adotadas:  $F_1$ : *Shifted Sphere Function*,  $F_2$ : *Schwefel's Problem 2.21*,  $F_3$ : *Shifted Rosenbrocks's Function*,  $F_4$ : *Shifted Rastrigin's Function*,  $F_5$ : *Shifted Griewank's Function* e  $F_6$ : *Shifted Ackley's Function*. Todas as funções foram executadas com 100 dimensões e são definidas como problemas de minimização.

Para avaliar o desempenho de nossas propostas para resolver problemas com mais de um objetivo, empregamos o conjunto de problemas da família de *benchmarks* DTLZ [Deb et al., 2002b]. Esta coleção de problemas possui as principais características necessárias a uma suíte de testes para avaliar adequadamente os MOEAs. Entre tais características, podemos destacar: simplicidade de concepção, escalabilidade em termos de número de variáveis de decisão ( $n$ ) e número de objetivos ( $M$ ), facilidade de compreensão da aparência da fronteira de Pareto resultante e dificuldade ajustável para se obter a fronteira de Pareto real.

O conjunto de *benchmarks* DTLZ é composto de sete problemas com propriedades distintas. Para cada problema, a fronteira de Pareto real é conhecida e a complexidade da busca é expressa pela variável  $k$ . O número total de variáveis é  $n = M + k - 1$ ,

onde  $M$  é o número de objetivos. Os problemas DTLZ são funções de minimização na forma  $\min f(\vec{X})$ . Nestes problemas,  $\vec{X}_M$  é um vetor de tamanho  $|\vec{X}_M| = k$  que contém os últimos valores de  $\vec{X}$ . Através de indicadores de qualidade é possível analisar os resultados gerados pelos MOEAs em termos de diversidade e convergência.

A descrição completa dos problemas de *benchmarks* utilizados nesse trabalho é exposta no Anexo A.

### 6.1.3 Testes estatísticos

Como previamente exposto, a avaliação da qualidade dos algoritmos evolutivos envolve a obtenção de indicadores de convergência e diversidade através de métricas de desempenho. Devido a diversos fatores, novos algoritmos evolutivos e variações de algoritmos já existentes têm sido propostos. Neste contexto, torna-se imprescindível que os resultados obtidos sejam corretos e rigorosamente analisados estatisticamente. Os testes estatísticos (ou de significância) são utilizados sobre os indicadores obtidos e servem para comparar os resultados das execuções dos algoritmos.

Segundo Coello *et al.* [Coello et al., 2007], o tipo de teste utilizado para avaliar o desempenho dos MOEAs é não-paramétrico, porque a distribuição da população para as métricas dos resultados é desconhecida. Além disso, esta distribuição não segue uma distribuição normal. Basicamente, testes estatísticos são utilizados na forma de testes de hipóteses. Uma hipótese estatística é uma suposição que pode ser verdadeira ou falsa [Demšar, 2006]. Por exemplo, uma suposição pode afirmar que o MOEA 1 é equivalente que o MOEA 2 em termos de média ou mediana. Este tipo de teste é dependente de uma probabilidade, chamada *p-value*, que serve para sustentar uma hipótese. O nível de significância é definido por  $\alpha$  que representa o maior *p-value* aceitável. Se o *p-value* gerado pelo teste estatístico é maior que  $\alpha$ , então a hipótese de nulidade é rejeitada, ou seja uma hipótese alternativa é aceita (os resultados são estatisticamente diferentes). Em relação à avaliação dos MOEAs, é desejável utilizar esta abordagem para a avaliação de hipóteses.

Existem muitos testes estatísticos para os indicadores de qualidade que podem

utilizados para comparar se dois ou mais MOEAs são diferentes entre si (melhor ou pior). Neste trabalho, são utilizados dois testes não paramétricos típicos [Coello et al., 2007, García et al., 2009]:

- **Friedman test** [Demšar, 2006]: o teste de Friedman é um teste não paramétrico utilizado para avaliar as diferenças entre várias amostras relacionadas. A hipótese nula para o teste de Friedman é de que não existem diferenças entre as amostras. Se a probabilidade *p-value* for menor que um nível de significância determinado, a hipótese nula é rejeitada. Assim, pode-se concluir que pelo menos duas das amostras são significativamente diferente uma das outras. No teste de Friedman, comparações múltiplas podem ser realizadas e quando a hipótese nula é rejeitada, ao menos um dos grupos é diferente dos demais. Porém, a informação de quais grupos são diferentes não está disponível. Assim, o procedimento de comparações múltiplas permite determinar quais pares de grupos são diferentes entre si.
- **Kruskal-Wallis** [García et al., 2009]: o teste Kruskal-Wallis (KWtest) é um método estatístico não paramétrico usado quando o tipo da distribuição não é conhecido. O KWtest é empregado para avaliar a hipótese nula de que todas as populações possuem as mesmas funções de distribuição em relação à hipótese alternativa de que ao menos duas das populações possuem funções de distribuição diferentes. Este teste estatístico pode ser realizado para cada indicador de qualidade utilizando os dados experimentais obtidos. Este teste é o principal método estatístico utilizado para determinar se duas amostras são da mesma população.

Os testes estatísticos apresentados empregam procedimentos para detectar diferenças significativas entre os resultados obtidos por diferentes algoritmos [Demšar, 2006]. Tais testes servem para afirmar que as diferenças observadas nos dados não são casuais [Derrac et al., 2011].

## 6.2 A otimização mono-objetivo com múltiplos enxames em várias topologias

Nesta seção, analisamos os resultados experimentais obtidos pela otimização com múltiplos enxames em várias topologias para a resolução de problemas mono-objetivo. Os experimentos foram inicialmente conduzidos considerando um conjunto de funções de *benchmarks* executados em um ambiente centralizado que executou o algoritmo PSO padrão. Para cada função de otimização, o processo de otimização (PSO padrão) foi individualmente executado e valores de referência foram gerados. Em seguida, cada função de otimização foi executada em um ambiente simulado usando múltiplos enxames e organizados em várias topologias, anteriormente explicadas. Resultados experimentais incluem o número de iterações e os passos de comunicação requeridos no procedimento de otimização. Os resultados da otimização centralizada e das diversas abordagens com múltiplos enxames foram estatisticamente analisados e são discutidos no decorrer desta Seção.

### 6.2.1 Simulação: metodologia & parâmetros

Como anteriormente exposto, executamos o algoritmo PSO utilizando um ambiente com múltiplos enxames distribuídos que foi simulado utilizando a ferramenta *SiMulation Programming Language* (SMPL) [MacDougall, 1987].

Através do SMPL, são alocados eventos de simulação em que cada enxame executa uma iteração no procedimento do PSO para refinar seu resultado. Se um enxame melhora seus resultados, então um evento de comunicação é iniciado. De acordo com a topologia, o enxame envia seu resultado a outro enxame. O enxame que recebe a informação compara o resultado recebido com seus próprios resultados. Se o resultado recebido é melhor, então a partícula recebida torna-se o *gbest* do enxame. Caso contrário, a informação é ignorada. Esse mecanismo garante que o compartilhamento do componente social entre os enxames ocorra de maneira assíncrona. A comunicação é iniciada por um evento (neste caso, a melhora de resultados) e não através

de um número pré-determinado de iterações.

A simulação termina quando o resultado para cada função apresenta um erro mínimo ou quando um número máximo (1.000.000) de iterações é atingido. A sobrecarga de comunicação é avaliada para cada topologia. O número de iterações, o número de mensagens utilizadas, assim como os resultados foram armazenadas a cada execução.

## Parâmetros de simulação utilizados

Para validar a proposta, o algoritmo PSO padrão [Eberhart e Shi, 2000] foi utilizado para gerar valores referenciais para serem comparados aos obtidos por nossa solução. Estes valores compreendem aos resultados obtidos das funções de otimização em um computador isolado (execução centralizada). Para a obtenção dos valores de referência, cada função utilizou 1000 partículas, cada uma composta por 100 valores inicializados aleatoriamente dentro da dimensão de cada função. A quantidade de partículas foi definida de modo a manter um equilíbrio quando da execução no ambiente distribuído. Os valores utilizados para os parâmetros são *default* [Eberhart e Shi, 2000]. O peso da inércia ( $\omega$ ) foi iniciado em 0,9 sendo gradativamente reduzido até um limite de 0,4. Esta opção se justifica, pois, ao se diminuir o peso da inércia, o resultado tende a ser refinado. As constantes  $C1$  e  $C2$  foram fixadas em 2,0, valores comumente empregados em experimentos na literatura pesquisada.

Cada partícula teve seus valores confinados nos limites de cada dimensão. Para as funções *Sphere*, *Griewank* e *Ackley*, cada execução foi realizada até que o melhor resultado atingisse um erro menor que 0,001 em relação ao ótimo global de cada função. Para as demais funções, o critério de parada foi um erro menor que 0,1. Para todas as funções foram realizadas 30 execuções e o melhor resultado e o número total de mensagens transmitidas de cada execução foi armazenado. Quanto menor a quantidade de iterações e mensagens trocadas, melhor o resultado. Caso a rodada não atinja o erro mínimo ao término de 1.000.000 de iterações, considerada-se que a execução não convergiu para o ótimo global. Essa situação também é registrada.



Na simulação distribuída, os mesmos parâmetros foram utilizados, excetuando-se o número de partículas associado a cada enxame. A totalidade de 1000 partículas foi uniformemente distribuída entre os enxames. Foram utilizados 16 enxames independentes entre si, cada um com 62 a 63 partículas. Esse cuidado foi tomado para que o custo computacional se mantivesse no mesmo patamar da execução de referência.

Conforme descrito anteriormente, foram utilizadas as topologias *broadcast*, anel unidirecional, anel bidirecional, *gossip* para um vizinho, *gossip* para  $\log(N)$  vizinhos, hipercubo e uma topologia dinâmica. A respeito da topologia dinâmica, ela inicia como uma topologia de grafo completo com  $\frac{N(N-1)}{2}$  arestas, e após 30.000 iterações ela torna-se uma topologia em anel com  $N$  arestas. Nesse trabalho,  $N = 16$ . O número total de arestas que precisam ser eliminadas é  $\frac{N(N-3)}{2}$  arestas. Assim, a cada  $\frac{30000}{(N-3)}$  iterações,  $m$  arestas são removidas, onde  $m$  segue um padrão de regressão aritmético na forma de  $N - 2, N - 3, \dots, 2$ .  $m$  arestas são removidas, uma por subpopulação. A aresta a ser removida é escolhida aleatoriamente entre as arestas que não pertençam ao anel exterior. A transformação de uma topologia inicial, totalmente conectada em uma topologia em anel é realizada em  $N - 3$  passos de eliminação.

Um aspecto adicional incorporado à simulação distribuída foi o troca do *gbest*. Ao contrário da máquina de referência que compartilha endereçamentos de memória de maneira fortemente acoplada, é necessário um mecanismo de comunicação explícito de troca do componente social entre os enxames. De acordo com a topologia entre os enxames, cada enxame escolhe um ou mais vizinhos e executa o processo descrito anteriormente. A troca do *gbest* é realizada por um enxame sempre que o mesmo é melhorado. Este mecanismo garante um método de compartilhamento assíncrono, evitando que comunicações ocorram desnecessariamente como observado no modelo síncrono que troca informações em uma frequência de iterações pré-fixadas.

Para cada modelo de topologia foram executadas 30 rodadas. A próxima seção apresenta os resultados obtidos e a discussão dos mesmos.

## 6.2.2 Resultados

Os resultados a seguir, apresentados na tabela 6.1, correspondem à média e ao desvio padrão (DP) dos 30 valores obtidos em cada uma das rodadas. Esses valores compreendem ao número de iterações necessárias para se atingir um resultado dentro da expectativa de erro definida. São apresentados os resultados de todas as funções em todas as topologias e estratégias de comunicação utilizadas. A tabela 6.2 apresenta a quantidade de mensagens enviadas nos modelos com múltiplos enxames.

Tabela 6.1: Resultados para cada função considerando diferentes topologias.

Topologia		Sphere	Schwefel 2,21	Rosenbrock
Referência	Média	<b>29032,63</b>	24	2
	DP	<b>736,30</b>	não convergiram	não convergiram
Broadcast	Média	<b>35243,47</b>	4	<b>171623,07</b>
	DP	<b>913,26</b>	não convergiram	<b>14002,91</b>
Dinâmica	Média	45856,90	<b>155003,90</b>	<b>175149,43</b>
	DP	1146,91	<b>12067,70</b>	<b>15072,77</b>
Hipercubo	Média	41748,60	165302,60	<b>171973,73</b>
	DP	1160,68	13754,06	<b>14531,19</b>
Gossip Log(N)	Média	60653,67	164329,17	<b>169705,43</b>
	DP	34729,88	10640,63	<b>17661,80</b>
Anel Bidirecional	Média	49326,90	<b>157296,03</b>	182138,73
	DP	1015,87	<b>9972,80</b>	12178,46
Anel Unidirecional	Média	51695,77	<b>156820,80</b>	185368,70
	DP	2270,94	<b>10579,48</b>	11114,19
Gossip 1	Média	47358,03	<b>157424,77</b>	<b>168320,30</b>
	DP	957,82	<b>10777,95</b>	<b>15718,40</b>
Topologia		Rastrigin	Griewank	Ackley
Referência	Média	3	4	<b>53154,23</b>
	DP	não convergiram	não convergiram	<b>1902,11</b>
Broadcast	Média	<b>110530,30</b>	<b>41369,17</b>	64558,27
	DP	<b>7410,34</b>	<b>1187,07</b>	2199,58
Dinâmica	Média	122454,50	3	73823,73
	DP	10316,42	não convergiram	2309,50
Hipercubo	Média	<b>115777,20</b>	5	66448,17
	DP	<b>8356,63</b>	não convergiram	1819,39
Gossip Log(N)	Média	<b>113740,40</b>	5	65535,33
	DP	<b>8166,78</b>	não convergiram	1932,66
Anel Bidirecional	Média	3	3	70804,30
	DP	não convergiram	não convergiram	2387,16
Anel Unidirecional	Média	4	4	74129,33
	DP	não convergiram	não convergiram	3121,07
Gossip 1	Média	1	4	69404,40
	DP	não convergiu	não convergiram	1551,86

Tabela 6.2: Custos de comunicação.

Topologia		Sphere	Schwefel 2,21	Rosenbrock
Broadcast	Média	32596,00	4	1052594,50
	DP	1334,85	não convergiram	209003,29
Dinâmica	Média	21301,37	194183,13	432354,90
	DP	596,22	38595,75	97950,53
Hiper cubo	Média	17106,13	345904,40	613873,33
	DP	555,48	69597,62	141762,07
Gossip Log(N)	Média	15716,67	331350,00	577488,13
	DP	439,14	53906,51	156441,61
Anel Bidirecional	Média	<b>8087,27</b>	<b>117656,33</b>	296907,07
	DP	<b>256,34</b>	<b>18527,19</b>	59810,58
Anel Unidirecional	Média	9518,47	<b>136610,80</b>	339162,37
	DP	280,36	<b>25730,57</b>	56223,00
Gossip 1	Média	<b>6965,20</b>	<b>114750,97</b>	<b>227063,27</b>
	DP	<b>267,03</b>	<b>21286,00</b>	<b>62592,10</b>
Topologia		Rastrigin	Griewank	Ackley
Broadcast	Média	2949667,00	<b>39646,50</b>	103207,50
	DP	3958764,01	<b>1613,43</b>	8002,25
Dinâmica	Média	<b>89871,00</b>	5	42089,87
	DP	<b>14560,72</b>	não convergiram	3002,75
Hiper cubo	Média	1273676,13	5	44785,73
	DP	1677997,76	não convergiram	3251,96
Gossip Log(N)	Média	1205075,20	5	42109,20
	DP	1588557,85	não convergiram	3120,67
Anel Bidirecional	Média	3	3	<b>17559,23</b>
	DP	não convergiram	não convergiram	<b>1790,63</b>
Anel Unidirecional	Média	4	4	21027,30
	DP	não convergiram	não convergiram	2928,17
Gossip 1	Média	1	4	<b>15582,77</b>
	DP	não convergiu	não convergiram	<b>1159,16</b>

Para comparar os resultados, foi realizado o teste de Friedman através do *R Project for Statistical Computing* [R Development Core Team, 2011]. O nível de significância considerado para o teste foi estabelecido em 5%. Quando o teste apresenta diferença estatisticamente significativa, as melhores médias são destacadas em negrito.

Para a função *Sphere*, o algoritmo PSO padrão apresenta os melhores resultados, porém, sem diferença estatística com o modelo de distribuição do componente social baseado em *broadcast*. Para a função de *Ackley* ocorre um comportamento similar, explicado pelo fato das duas funções terem seus resultados influenciados fortemente pela posição da melhor partícula. Para todas as outras funções, o desempenho do algoritmo PSO padrão deixou a desejar, pois não conseguiu convergir para os resul-

tados desejados dentro de um limite de 1.000.000 de iterações. Considerando, por exemplo, a função de *Schwefel 2.21*, das 30 rodadas, em 24 ocasiões, o algoritmo PSO padrão não conseguiu atingir o resultado esperado. Isso é explicado pelo fato de o algoritmo PSO padrão estagnar em ótimos locais, não podendo “fugir” para outras regiões.

Porém, é possível observar que todos os modelos ocasionalmente apresentam problemas de convergência. A função de *Griewank* é uma função que conduz facilmente a ótimos locais e em todos os modelos, à exceção do *broadcast*, algumas rodadas que não convergiram ao resultado desejado. Nessa situação, observa-se que o mecanismo de comunicação assíncrono ajudou o modelo *broadcast* a convergir seus resultados em todas as rodadas. Como o componente social não foi compartilhado em intervalos fixos, este mecanismo teve mais sucesso em evitar ótimos locais. O ponto negativo do mecanismo de *broadcast* é o excessivo custo de comunicação, porém para algumas funções, a topologia teve influência ao apresentar ótimos resultados a um custo de comunicação relativamente baixo.

Para a função de *Rosenbrock*, a disseminação epidêmica (*gossip*) para um vizinho necessitou de menos iterações para atingir o resultado desejado. Este bom desempenho é significativo, pois foi o mecanismo que realizou menos comunicações. Para a função de *Rastrigin*, a topologia *broadcast* necessitou de menos iterações, embora a um custo de comunicação consideravelmente superior que a topologia dinâmica.

A Figura 6.1 ilustra a comunicação realizada, incluindo o número de enxames que enviam seu componente social (*senders*) e o número de mensagens enviadas (*messages*). À esquerda são ilustrados aspectos de comunicação para a problema *Schwefel 2.21* usando a topologia dinâmica enquanto que à direita da Figura 6.1 é ilustrado o comportamento da função de *Rosenbrock* utilizando a topologia *gossip* para um vizinho. Aqui são mostrados os resultados de uma rodada. O eixo horizontal *x* representa o número de iterações (tempo) e o eixo vertical *y* representa a quantidade de comunicação realizada. Como anteriormente mencionado, uma comunicação ocorre quando um enxame melhora seu próprio resultado e o envia para outro(s) enxame(s).

Na Figura 6.1 (esquerda) é possível observar que menos que 25% das mensagens são transmitidas durante a primeira metade de toda a otimização (77501 iterações). A diminuição da conectividade ajudou a economizar mensagens e melhorar o tempo requerido para obter os resultados. Na Figura 6.1 (direita) é possível observar que uma melhora significativa ocorre após a primeira metade da otimização (90.000 iterações). O pequeno número de vizinhos é, sem dúvidas, a razão para os bons resultados produzidos pela execução da função de *Rosenbrock* na topologia *gossip* para um vizinho.

De modo a ilustrar situações em que a convergência não é atingida, são consideradas duas rodadas da função de *Griewank* na topologia dinâmica. É possível observar que o algoritmo deve escapar do ótimo local nas primeiras iterações. Quando a função fica estagnada em um ótimo local, o número de iterações não ajuda a reduzir o tempo de convergência.

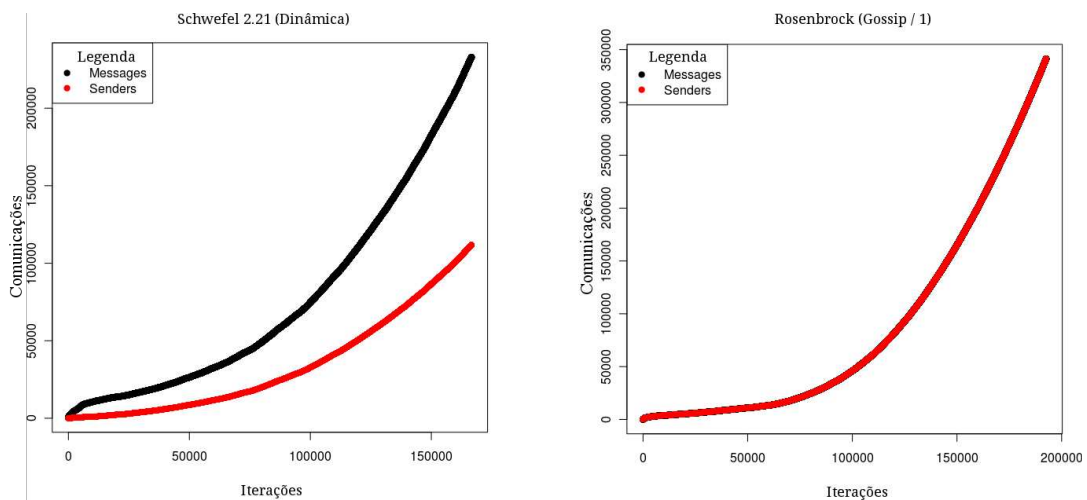


Figura 6.1: Custo de comunicação para as Funções *Schwefel's 2.21* e *Rosenbrock*.

Como ilustrado pela Figura 6.2 (direita) não há melhora nos resultados entre 160.000 e 1.000.000 de iterações. A Figura 6.2 (esquerda) ilustra uma situação onde a mesma função utilizando a mesma topologia obteve resultados que convergiram para o esperado.

Em relação à topologia hipercubo, é possível observar que para apenas um problema não foi possível convergir o resultado em todas as rodadas. Porém há de se

destacar a influência da desejável propriedade logarítmica da referida topologia. Considerando a disposição dos enxames na forma de um grafo, a partir de um determinado enxame, o enxame mais distante deste está uma distância máxima de  $\log(N)$ . Isto é, ao se dobrar o número de enxame, a distância máxima entre dois enxames aumenta em apenas um. Esta propriedade é especialmente importante para resolução de problemas complexos, onde a quantidade elevada de enxames é necessária.

Uma característica observada em todas as topologias e para a maioria das funções é que no início da otimização, o *gbest* é facilmente melhorado e assim, uma quantidade elevada de mensagens é trocada entre os enxames. Todavia, à medida que as iterações aumentam, esta melhora ocorre com menor frequência. Existe uma tendência de que os resultados das funções estabilizem. No modelo de comunicação síncrono, como a comunicação acontece em intervalos fixos, pode ocorrer que um enxame compartilhe desnecessariamente uma partícula ruim. Nesta situação, o modelo de comunicação assíncrono mostra-se vantajoso, pois a comunicação ocorre apenas quando há uma melhora no *gbest*.

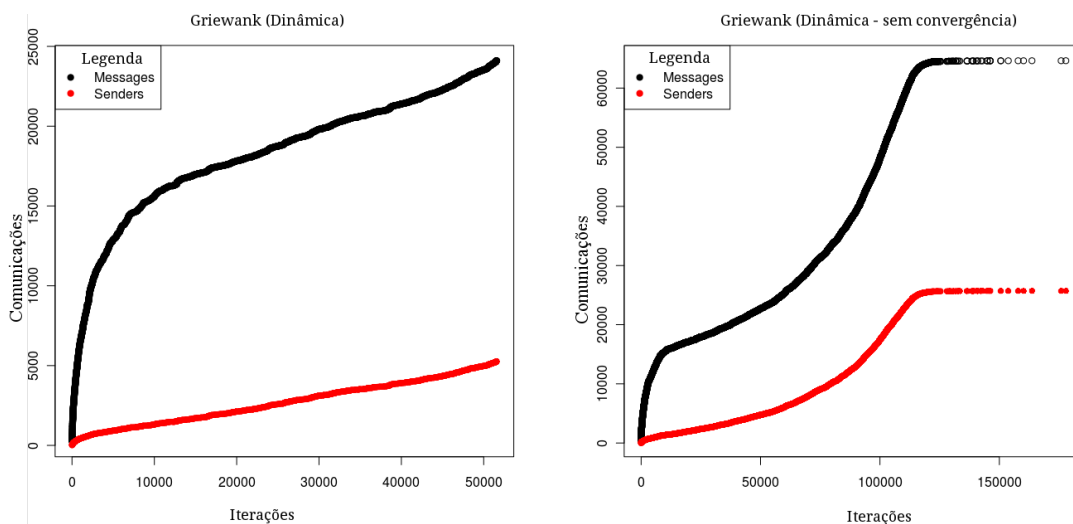


Figura 6.2: Custo de comunicação para a Função *Griewank* na topologia dinâmica.

Para entender esta relação, considere a função *Sphere*. Na maioria das abordagens com múltiplos enxames com comunicação síncrona, ocorre uma comunicação por iteração. Na topologia *gossip* para um vizinho, por exemplo, foram necessárias, em média, cerca de 47.358,03 iterações e 6665,20 comunicações para alcançar o

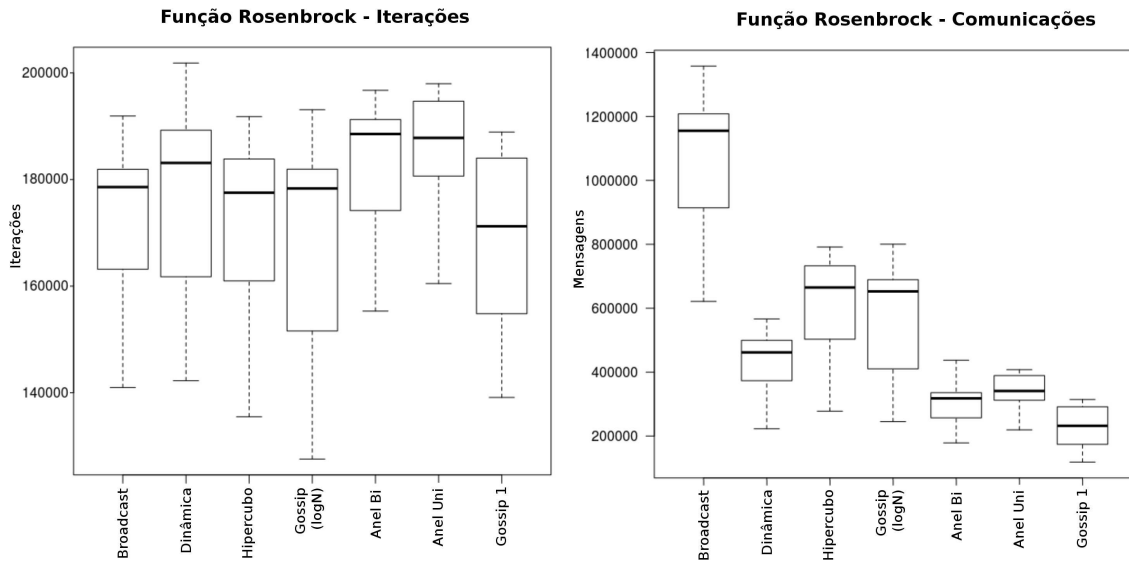


Figura 6.3: Diagrama *Box-and-whisker* para a função de *Rosenbrock*: convergência e comunicação, respectivamente.

resultado. Isto é, uma comunicação a cada 6.8 iterações.

É possível concluir que, para algumas funções, a baixa conectividade exerceu uma influência positiva nos resultados. Para isso, basta observar os diagramas *box-and-whisker* (boxplots) da Função de *Rosenbrock*, ilustrado na Figura 6.3. É possível observar que, além de ajudar a economizar mensagens (diagrama à direita), a topologia gossip para um vizinho apresentou os melhores resultados entre todas as topologias (diagrama à esquerda).

Para avaliar o desempenho individual de cada topologia, aplicamos o critério de avaliação usado em [Suganthan et al., 2005] chamado de **Success Performance** (SP). Este critério favorece as topologias que apresentem mais execuções bem sucedidas, ou seja, rodadas com resultados que convirjam para o esperado. De modo a ordenar as topologias da melhor para a pior, calculamos o SP para cada problema. O SP de cada topologia =  $\text{média}(\text{rodadas bem sucedidas}) * \frac{(\#total \text{ de rodadas})}{(\#rodadas \text{ bem sucedidas})}$ . As expressões são computadas para cada problema separadamente. A seguir, o SP de cada topologia é dividido pelo SP do melhor algoritmo para cada função. Assim, a melhor topologia para cada função terá o valor 1. À medida que esse valor aumenta, um ranqueamento é estabelecido da melhor para a pior topologia. Assim, baseado no SP, podemos ordenar as topologias. A tabela 6.3 mostra este ranqueamento.

Tabela 6.3: Ranqueamento das topologias baseado no *Success Performance*.

Topologia	Sphere	Schwefel2.21	Rosenbrock	Rastrigin	Griewank	Ackley
Broadcast	1. <sup>o</sup>	7. <sup>o</sup>	4. <sup>o</sup>	1. <sup>o</sup>	1. <sup>o</sup>	2. <sup>o</sup>
Dinâmica	5. <sup>o</sup>	2. <sup>o</sup>	6. <sup>o</sup>	5. <sup>o</sup>	4. <sup>o</sup>	7. <sup>o</sup>
Hipercubo	3. <sup>o</sup>	6. <sup>o</sup>	2. <sup>o</sup>	2. <sup>o</sup>	3. <sup>o</sup>	3. <sup>o</sup>
Gossip <sub>log(n)</sub>	2. <sup>o</sup>	5. <sup>o</sup>	3. <sup>o</sup>	3. <sup>o</sup>	2. <sup>o</sup>	1. <sup>o</sup>
Anel Bidir.	6. <sup>o</sup>	3. <sup>o</sup>	5. <sup>o</sup>	6. <sup>o</sup>	5. <sup>o</sup>	5. <sup>o</sup>
Anel Uni.	7. <sup>o</sup>	4. <sup>o</sup>	7. <sup>o</sup>	7. <sup>o</sup>	7. <sup>o</sup>	6. <sup>o</sup>
Gossip 1	4. <sup>o</sup>	1. <sup>o</sup>	1. <sup>o</sup>	4. <sup>o</sup>	6. <sup>o</sup>	4. <sup>o</sup>

O teste de Friedman foi aplicado nos resultados apresentados pelo SP e não acusou diferenças entre as topologias. Por exemplo, a topologia *broadcast* foi a melhor para a função *Sphere* e a pior para o problema *Schwefel's 2.21*. Em outras palavras, a melhor topologia para uma função pode ser a pior topologia para outra função. Assim, é recomendado avaliar qual topologia é melhor para uma função em específico.

A topologias *broadcast* e *gossip* apresentaram os melhores resultados em termos de convergência. Outra conclusão é que é uma boa estratégia iniciar com uma topologia altamente conectada e depois diminuir a conectividade quando a melhora do componente social torna-se um evento mais raro.

### 6.2.3 Influência do número de enxames

Neste trabalho, avaliamos também o impacto no desempenho do processo de otimização quando o número de subpopulações é diminuído ou aumentado. Inicialmente duas funções foram escolhidas para serem otimizadas: uma considerada “fácil”, no caso a *Sphere* e outra considerada “difícil” (*Rosenbrock*). Esse nível de dificuldade foi estabelecido em função do número de iterações necessária para que os algoritmos convergissem para os resultados estabelecidos. Menos iterações necessárias para o *benchmark* considerado “fácil” (mais iterações para o “difícil”). Tais funções foram executadas com 8, 16 e 32 enxames e cada função foi executada 30 vezes em cada topologia. Os melhores resultados obtidos foram armazenados após 60.000 iterações.

A Tabela 6.4 mostra a média (M) e o desvio padrão (dp) dos resultados obtidos para cada função e para cada topologia com 8, 16 e 32 enxames.



Tabela 6.4: Desempenho dos múltiplos enxames com 8, 16 e 32 subpopulações para as funções *Sphere* e *Rosenbrock*.

Topologia		Sphere			Rosenbrock		
		8 Nodos	16 Nodos	32 Nodos	8 Nodos	16 Nodos	32 Nodos
Broadcast	M	<b>0,000004</b>	0,000004	0,000005	<b>186,97</b>	208,09	231,12
	dp	<b>0,000001</b>	0,000001	0,000001	<b>99,938</b>	166,75	239,77
Dinâmica	M	<b>0,000013</b>	0,000053	0,000337	<b>263,35</b>	386,79	449,06
	dp	<b>0,000003</b>	0,000011	0,000061	<b>209,47</b>	397,68	387,35
Hipercubo	M	<b>0,000002</b>	0,000005	0,000015	<b>163,93</b>	334,40	323,04
	dp	<b>0,000009</b>	0,000020	0,000052	<b>212,36</b>	282,14	254,82
Gossip Log(N)	M	<b>0,000063</b>	0,000012	0,000005	<b>189,83</b>	297,76	284,83
	dp	<b>0,000013</b>	0,000003	0,000001	<b>55,25</b>	386,98	297,25
Anel Bidir.	M	<b>0,000024</b>	0,000122	0,000908	<b>307,45</b>	360,17	237,50
	dp	<b>0,000005</b>	0,000028	0,000146	<b>333,66</b>	396,21	197,69
Anel Unidir.	M	<b>0,000036</b>	0,000205	0,003033	<b>253,84</b>	360,61	398,19
	dp	<b>0,000008</b>	0,000031	0,000422	<b>227,78</b>	405,52	353,20
Gossip 1	M	<b>0,000020</b>	0,000072	0,000305	<b>222,03</b>	212,79	495,26
	dp	<b>0,000006</b>	0,000013	0,000072	<b>175,21</b>	222,35	432,73

Para comparar os resultados, o teste Kruskal-Wallis foi aplicado com nível de significância igual a 5% através do *R Project for Statistical Computing* [R Development Core Team, 2011]. Para instâncias que, de acordo com o teste, apresentaram diferença estatística, os melhores resultados foram destacados em negrito. Os resultados para cada topologia foram analisados de acordo com o número de enxames. O teste mostrou que, para ambas funções, a configuração com 8 enxames apresentou os melhores resultados. É possível concluir que manter poucos enxames com muitas partículas foi uma boa estratégia para a maioria dos casos. Porém, é importante ressaltar que, assim como não existe uma única topologia que apresente os melhores resultados para todas as funções, não existe um único número de enxames que seja adequado para todas as funções.

A partir dos experimentos apresentados neste capítulo, é possível concluir que a execução de múltiplos enxames em qualquer topologia avaliada apresenta um efeito positivo na convergência do processo de otimização. Isto ocorre pela melhor exploração do espaço de busca por subpopulação separadas e independentes que se comunicam quando ocorre alguma melhora em seus resultados. Para funções simples, a alta conectividade melhora a convergência, entretanto, para funções complexas, esta

alta conectividade pode conduzir os resultados para ótimos locais. Em alguns casos, as topologias com baixa conectividade foram mais rápidas e empregaram menos comunicação. Para algumas situações, bons resultados foram obtidos ao se iniciar a otimização com uma topologia altamente conectada e depois diminuir a conectividade quando a melhora do componente social torna-se um evento mais raro.

Para avaliar o desempenho individual de cada topologia, um ranqueamento foi estabelecido e foi possível observar que uma topologia que apresenta bons resultados para uma função pode apresentar um péssimo desempenho para outra função.

Variamos o número de enxames para investigar seu efeito na otimização distribuída. Funções simples e complexas foram avaliadas e para ambos os tipos de funções, a configuração com 8 enxames apresentou os melhores resultados: na maioria dos casos, poucos enxames com mais partículas são efetivamente melhores.

### **6.3 A otimização com múltiplos enxames e baseada em conceitos de dominância para resolver problemas com vários objetivos**

Nesta seção descrevemos a avaliação empírica do MSPSO proposto para resolver problemas com vários objetivos. Utilizamos dois *benchmarks* tradicionais da família DTLZ [Deb et al., 2002b]: DTLZ2 e DTLZ4. No problema DTLZ2, a fronteira de Pareto ótima corresponde ao primeiro quadrante de uma unidade esférica. A DTLZ4 é uma modificação da DTLZ2, porém com uma equação paramétrica diferente que gera um conjunto denso de soluções no plano  $f_M - f_1$ . Para cada problema, se conhece a fronteira de Pareto real e a complexidade da busca é expressada por  $k$ . O número total de variáveis é  $n = M + k - 1$ , onde  $M$  é o número de objetivos. A função DTLZ2 foi adotada porque permite a investigação da escalabilidade dos algoritmos à medida que o número de objetivos aumenta. A função DTLZ4 foi usada para investigar o comportamento de nosso MOEA para manter uma boa distribuição de soluções.

Os experimentos foram executados considerando problemas com 2, 3, 4, 5, 10, 15 e 20 objetivos. Uma versão sequencial do algoritmo 9 (MOPSO Usando o valor  $\sigma$  com

*Crowding Distance*) foi inicialmente executada para gerar resultados de referência. Em seguida, os problemas foram executados em ambas as versões distribuídas. A otimização termina quando um número pré-definido de 100 iterações é atingido. O mesmo critério de parada foi adotado para as execuções de referência.

### 6.3.1 Simulação: metodologia & parâmetros

Novamente, implementamos nossa proposta através da linguagem de programação C utilizando a ferramenta de simulação baseada em eventos SMPL [MacDougall, 1987].

Como mencionado anteriormente, inicialmente avaliamos o algoritmo sequencial (Algoritmo 9) para gerar resultados referenciais. Para obter esses resultados, o algoritmo MOPSO utilizou 512 partículas, cada uma consistindo de  $n$  valores inicializados aleatoriamente no escopo de cada função. O número total de variáveis é  $n = M + k - 1$ , onde  $M$  é o número de objetivos; 2, 3, 4, 5, 10, 15 e 20 objetivos foram avaliados. Nos experimentos,  $k = 10$ . Os valores adotados como parâmetros são os tradicionalmente empregados na literatura [Eberhart e Shi, 2000]. O peso de inércia ( $w$ ) foi inicializado em 0,9 e foi gradativamente reduzido para 0,4. Optamos por este processo, pois reduzir o peso da inércia no processo iterativo tende a melhorar os resultados [Eberhart e Shi, 2000]. As constantes  $C1$  e  $C2$  foram fixadas em 2. O tamanho do repositório de soluções não dominadas foi definido em 512.

A mesma abordagem foi empregada para avaliar o algoritmo distribuído proposto, exceto que as 512 partículas foram distribuídas entre 8 enxames (64 partículas por enxame). Como concluído na Seção anterior e reportado em [de Campos et al., 2013], manter poucos enxames com muitas partículas é uma boa estratégia. O tamanho do repositório também foi definido em 64. Esta configuração permite uma comparação justa com os resultados de referência.

Uma característica adicional incorporada à simulação do algoritmo de otimização distribuído foi a troca de soluções não dominadas. De acordo com a topologia, cada enxame envia seu *buffer* (repositório) para outros vizinhos, como descrito anteriormente. Soluções não dominadas são trocadas quando elas são melhoradas. Esta

estratégia de comunicação assíncrona evita comunicações desnecessárias em comparação ao modelo síncrono (que estabelece que a informação é trocada periodicamente dentro de um número fixo de iterações, mesmo que nenhuma nova solução não dominada seja criada).

Todos os algoritmos considerados executaram 30 rodadas independentes para cada número de objetivos, e para cada execução, o conteúdo final do repositório ou conjunto de *gbest* foi armazenado. Para as métricas GDp, IGDp e *Spacing*, valores menores são melhores.

A seguir, discutimos os resultados de cada função de *benchmark* e comparamos os resultados obtidos pelas estratégias distribuídas com aqueles obtidos pela estratégia sequencial (referência). Ambas as Tabelas 6.5 e 6.6 mostram a média e o desvio padrão (DP) computado das amostras obtidas das 30 execuções de cada função. Para comparar os resultados, novamente utilizamos o teste de Friedman, o qual foi aplicado com nível de significância de 5% utilizando a ferramenta estatística *R toolkit for statistical computing* (<http://www.r-project.org/>) [R Development Core Team, 2011]. Os resultados foram destacados em negrito quando o teste acusou diferença estatística significativa.

### 6.3.2 Resultados

Os resultados apresentados na Tabela 6.5 foram obtidos das execuções sequencial e paralelas para a função DTLZ2. A Tabela 6.6 apresenta os valores correspondentes para a função DTLZ4. As duas tabelas mostram os resultados relativos às métricas GDp, IGDp e *Spacing*.

A métrica GDp indica que, para a função DTLZ2, a execução *broadcast* fornece os melhores resultados, com diferença estatística significativa em comparação com as abordagens centralizada (referência) e *gossip* (para 5 objetivos, *gossip* apresentou bom desempenho). Para a função DTLZ4 com 2, 3 e 4 objetivos, a métrica GDp indica que a estratégia *broadcast* supera as estratégias centralizada e *gossip*. Quando o número de objetivos varia de 5 a 20, embora não se observe diferença estatística entre

Tabela 6.5: Indicadores de desempenho do problema DTLZ2.

# de Objetivos		GDp			IGDp			Spacing		
		Ref.	Gossip	Broad.	Ref.	Gossip	Broad.	Ref.	Gossip	Broad.
2	Média	4,41E-1	4,46E-2	<b>1,47E-2</b>	1,25E-1	<b>8,45E-3</b>	<b>1,02E-2</b>	9,00E-1	<b>4,88E-1</b>	<b>4,59E-1</b>
	DP	1,93E-1	4,49E-2	<b>1,78E-2</b>	3,99E-2	<b>8,49E-3</b>	<b>4,86E-3</b>	1,48E-1	<b>4,88E-1</b>	<b>3,28E-2</b>
3	Média	5,82E-1	1,98E-1	<b>1,07E-1</b>	1,95E-1	<b>1,29E-1</b>	1,69E-1	1,14E+0	<b>7,69E-1</b>	<b>6,61E-1</b>
	DP	1,90E-1	1,99E-1	<b>4,84E-2</b>	5,57E-2	<b>1,29E-1</b>	7,41E-2	1,57E-1	<b>7,71E-1</b>	<b>1,03E-1</b>
4	Média	5,34E-1	2,99E-1	<b>2,00E-1</b>	2,20E-1	2,30E-1	2,26E-1	1,03E+0	8,73E-1	<b>7,84E-1</b>
	DP	1,74E-1	2,98E-1	<b>7,80E-2</b>	6,32E-2	2,30E-1	8,11E-2	1,09E-1	8,79E-1	<b>9,78E-2</b>
5	Média	5,96E-1	<b>3,39E-1</b>	<b>2,36E-1</b>	<b>2,69E-1</b>	<b>2,70E-1</b>	2,75E-1	1,06E+0	<b>8,90E-1</b>	<b>8,34E-1</b>
	DP	1,99E-1	<b>3,38E-1</b>	<b>7,31E-2</b>	<b>7,68E-2</b>	<b>2,70E-1</b>	8,44E-2	1,39E-1	<b>8,90E-1</b>	<b>1,12E-1</b>
10	Média	8,08E-1	5,16E-1	<b>3,52E-1</b>	3,94E-1	<b>3,43E-1</b>	<b>3,17E-1</b>	1,17E+0	<b>9,89E-1</b>	<b>9,16E-1</b>
	DP	2,65E-1	5,17E-1	<b>1,09E-1</b>	1,15E-1	<b>3,43E-1</b>	<b>9,02E-2</b>	1,48E-1	<b>9,88E-1</b>	<b>7,41E-2</b>
15	Média	9,37E-1	6,16E-1	<b>4,50E-1</b>	4,33E-1	<b>3,77E-1</b>	<b>3,34E-1</b>	1,23E+0	<b>1,05E+0</b>	<b>1,02E+0</b>
	DP	2,95E-1	6,12E-1	<b>1,34E-1</b>	1,38E-1	<b>3,76E-1</b>	<b>9,29E-2</b>	1,34E-1	<b>1,05E+0</b>	<b>1,24E-1</b>
20	Média	9,76E-1	6,89E-1	<b>4,46E-1</b>	4,14E-1	4,00E-1	<b>3,49E-1</b>	1,23E+0	1,07E+0	<b>9,99E-1</b>
	DP	3,20E-1	6,91E-1	<b>1,35E-1</b>	1,36E-1	4,00E-1	<b>9,86E-2</b>	1,49E-1	1,07E+0	<b>8,86E-2</b>

Tabela 6.6: Indicadores de desempenho do problema DTLZ4.

# de Objetivos		GDp			IGDp			Spacing		
		Ref.	Gossip	Broad.	Ref.	Gossip	Broad.	Ref.	Gossip	Broad.
2	Média	2,75E-1	<b>3,36E-2</b>	<b>1,83E-2</b>	7,75E-2	<b>8,51E-3</b>	<b>8,73E-3</b>	7,66E-1	4,67E-1	<b>4,18E-1</b>
	DP	1,14E-1	<b>3,43E-2</b>	<b>1,68E-2</b>	2,54E-2	<b>8,72E-3</b>	<b>7,03E-3</b>	9,18E-2	4,65E-1	<b>2,49E-2</b>
3	Média	<b>4,09E-1</b>	2,18E-1	<b>1,44E-1</b>	9,11E-2	<b>3,64E-2</b>	4,84E-2	1,05E+0	8,24E-1	<b>7,44E-1</b>
	DP	1,61E-1	2,18E-1	<b>5,18E-2</b>	3,14E-2	<b>3,63E-2</b>	2,02E-2	1,36E-1	8,25E-1	<b>6,40E-2</b>
4	Média	4,81E-1	4,82E-1	<b>3,14E-1</b>	1,40E-1	1,72E-1	1,45E-1	1,14E+0	<b>1,00E+0</b>	<b>8,96E-1</b>
	DP	1,56E-1	4,79E-1	<b>1,08E-1</b>	4,24E-2	1,72E-1	6,11E-2	6,52E-2	<b>1,00E+0</b>	<b>1,13E-1</b>
5	Média	4,90E-1	5,57E-1	<b>4,17E-1</b>	1,41E-1	1,79E-1	1,52E-1	1,09E+0	<b>9,89E-1</b>	<b>8,72E-1</b>
	DP	1,73E-1	5,58E-1	<b>1,51E-1</b>	4,46E-2	1,79E-1	5,32E-2	1,39E-1	<b>9,88E-1</b>	<b>1,18E-1</b>
10	Média	<b>6,45E-1</b>	8,06E-1	<b>6,55E-1</b>	3,10E-1	<b>3,22E-1</b>	<b>2,73E-1</b>	1,02E+0	1,09E+0	<b>9,50E-1</b>
	DP	<b>1,95E-1</b>	8,03E-1	<b>1,86E-1</b>	9,38E-2	<b>3,21E-1</b>	<b>7,61E-2</b>	7,64E-2	1,08E+0	<b>8,30E-2</b>
15	Média	7,50E-1	8,51E-1	7,46E-1	4,00E-1	4,06E-1	3,58E-1	1,04E+0	1,12E+0	<b>9,92E-1</b>
	DP	2,22E-1	8,56E-1	2,07E-1	1,16E-1	4,06E-1	1,00E-1	6,52E-2	1,12E+0	<b>7,57E-2</b>
20	Média	<b>7,31E-1</b>	8,76E-1	<b>7,80E-1</b>	4,54E-1	4,76E-1	4,24E-1	<b>1,05E+0</b>	1,15E+0	<b>1,06E+0</b>
	DP	<b>2,09E-1</b>	8,77E-1	<b>2,15E-1</b>	1,25E-1	4,75E-1	1,17E-1	<b>7,73E-2</b>	1,15E+0	<b>6,76E-2</b>

os resultados de referência e os obtidos pela estratégia *broadcast*, na maioria dos casos, o algoritmo com múltiplos enxames usando *broadcast* apresentou resultados melhores. A estratégia *gossip* apresentou os piores resultados, indicando que, para a função DTLZ4, um baixo nível de interação entre partículas prejudica a convergência.

Por outro lado, quando a métrica IGDp é considerada, é possível notar que para a função DTLZ2, a estratégia *gossip* teve boa influência na diversidade dos resultados. Fica claro que não é possível eleger uma única estratégia como a melhor, mas levando em conta o conjunto de resultados, pode-se observar que as estratégias distribuídas foram melhores que a alternativa centralizada. Apenas para 5 objetivos, a estratégia de referência apresenta resultados similares em comparação com a estratégia *gossip*. E mesmo neste caso, os resultados de referência foram piores. Para 4 objetivos, nenhuma diferença estatística pôde ser observada, embora a estratégia *gossip* apresente os melhores resultados. Para a função DTLZ4, a métrica IGDp indica que as estratégias *broadcast* e *gossip* fornecem os melhores resultados para dois objetivos. Para três objetivos, o algoritmo *gossip* mostra melhor desempenho, enquanto que as execuções *broadcast* mostram melhores resultados para 10 objetivos. Novamente, para este cenário, os resultados de referência não foram particularmente bons. Enquanto que para a função DTLZ2 em apenas uma situação não foi possível observar diferença estatística entre os resultados, para a função DTLZ4 isso aconteceu frequentemente: para 4, 5, 15 e 20 objetivos não foi possível constatar diferenças estatísticas. Para demonstrar melhor esta situação, a Figura 6.4 ilustra o diagrama *box-and-whisker* (boxplot) dos resultados obtidos. Na maioria dos casos, o algoritmo *broadcast* pode ser considerado a melhor opção.

Para avaliar a distribuição dos resultados ao longo das fronteiras, utilizamos a métrica *spacing*. Esta métrica mede a faixa de variação entre soluções vizinhas na fronteira e o valor 0 indica que todas as soluções estão igualmente distribuídas no espaço de objetivos. Para a função DTLZ2, a comunicação baseada em *broadcast* é a melhor escolha em termos de boa distribuição. Para 2, 3, 5, 10 e 15 objetivos, *gossip* apresentou bons resultados também. Em todos os casos, os testes estatís-

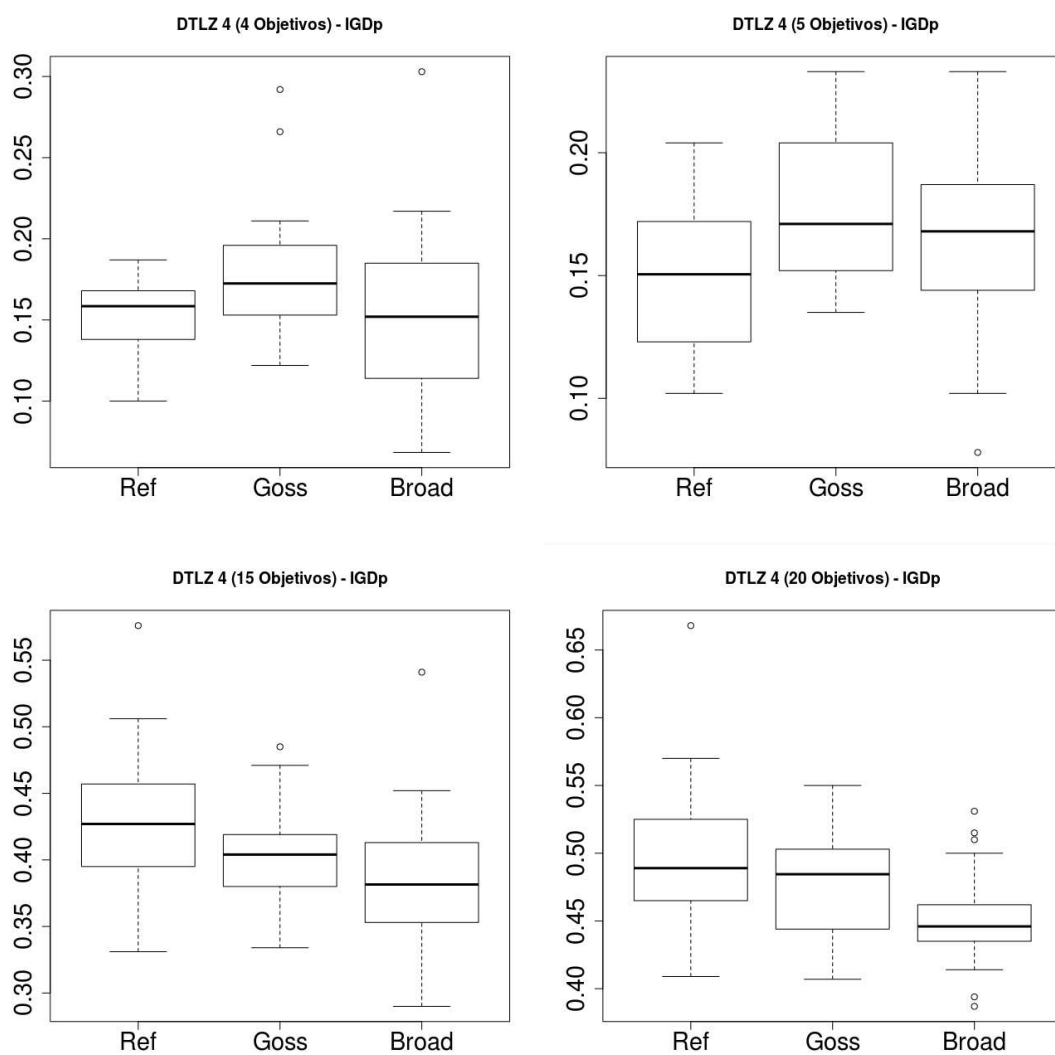


Figura 6.4: Boxplot dos resultados obtidos pela métrica IGDp para a função DTLZ4 que não apresentaram diferença estatística.

ticos indicam que o modelo centralizado apresenta um desempenho inferior. Para a função DTLZ4, um comportamento similar pode ser observado. Novamente, a estratégia *broadcast* apresentou desempenho superior. Uma exceção pode ser observada com 20 objetivos, na qual a execução referencial apresentou uma boa distribuição.

## Eficiência referente ao compartilhamento de soluções

A partir da observação dos resultados obtidos, é possível concluir que as alternativas com múltiplos enxames distribuídos fornece boas soluções para problemas multiobjetivo e com vários objetivos. Tendo isso em mente, decidimos investigar a efetividade do compartilhamento de soluções não dominadas entre os enxames. Como mencionado, quando um *buffer* é enviado para um enxame, o destino avalia as soluções recebidas. Cada solução recebida pode dominar, ser dominada ou não apresentar relação de dominância em comparação às soluções locais. Neste último caso, se o repositório não está cheio, a solução recebida é simplesmente adicionada ao repositório. Caso contrário, o método CD é aplicado e define se uma solução recebida deve substituir uma solução do repositório ou deve ser ignorada. Na prática, tais situações definem duas possibilidades: ou a solução é adicionada ao repositório destino ou a solução é ignorada. A seguir, apresentamos os resultados que mostram o quanto a estratégia de compartilhamento de repositório é boa em ambos os algoritmos. De forma a permitir a análise do compartilhamento para poucos e vários objetivos, apresentamos as situações com 2 e 20 objetivos. Para a estratégia de comunicação baseada em *broadcast*, um enxame envia seu *buffer* para todos seus vizinhos. As Figuras 6.5, 6.6, 6.7 e 6.8 mostram o número médio de soluções que foram adicionadas ao buffer e que foram ignoradas. Quando a estratégia baseada em *gossip* é adotada, apenas um vizinho é aleatoriamente escolhido como destino.

A Figura 6.5 ilustra uma amostra de padrão de comunicação obtido pela execução das estratégias *broadcast* e *gossip* para resolver a função DTLZ2 com dois objetivos. É possível perceber que, ao longo do tempo, a alta conectividade da topologia *broadcast* ajuda os enxames a gerar mais soluções não dominadas. Assim, o tamanho do



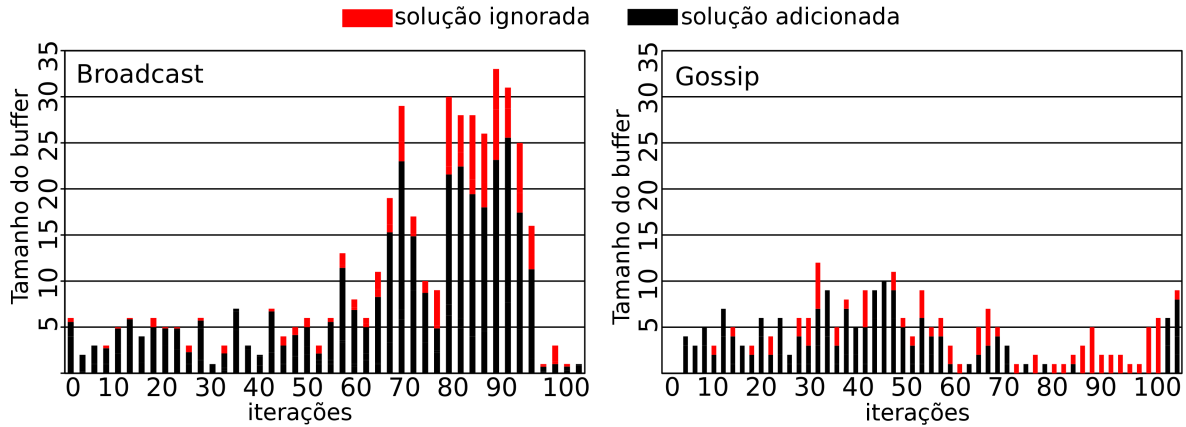


Figura 6.5: Padrão de comunicação para a função DTLZ2 (2 objetivos).

*buffer* aumenta ao longo do tempo. Considerando o padrão estabelecido pelo *gossip*, é possível notar a dificuldade para que o *buffer* aumente de tamanho ao longo do tempo. Isto é decorrente do baixo nível de interação entre enxames diferentes. Uma boa surpresa foi o fato de que na maioria dos casos a informação compartilhada era relevante e foi incorporada pelos enxames receptores.

A Figura 6.6 ilustra um padrão de comunicação para a estratégia *broadcast* e um padrão para a estratégia *gossip* para resolver a função DTLZ2 com vários objetivos. A partir das primeiras iterações, o tamanho do *buffer* é grande porque a maioria das soluções geradas são não dominadas devido ao elevado número de objetivos (20). Porém, à medida que o processo evolui, é possível observar que o tamanho do *buffer* diminui quando a estratégia *gossip* é considerada. Para a alternativa *broadcast*, o tamanho do *buffer* não diminui. Isto é, a baixa conectividade evita que os enxames criem novas soluções não dominadas. Assim, para vários objetivos, é possível argumentar que a execução distribuída com muita interatividade entre os enxames foi responsável pelos bons resultados obtidos pela estratégia *broadcast* para todos os indicadores mostrados na Tabela 6.5.

A Figura 6.7 ilustra o padrão de comunicação de uma amostra de execução da estratégia *broadcast* e uma amostra da estratégia *gossip* para resolver a função DTLZ4 com 2 objetivos. Em ambos os casos é possível observar que ao longo do tempo, o tamanho do *buffer* não atinge mais que cinco soluções não dominadas. Apesar

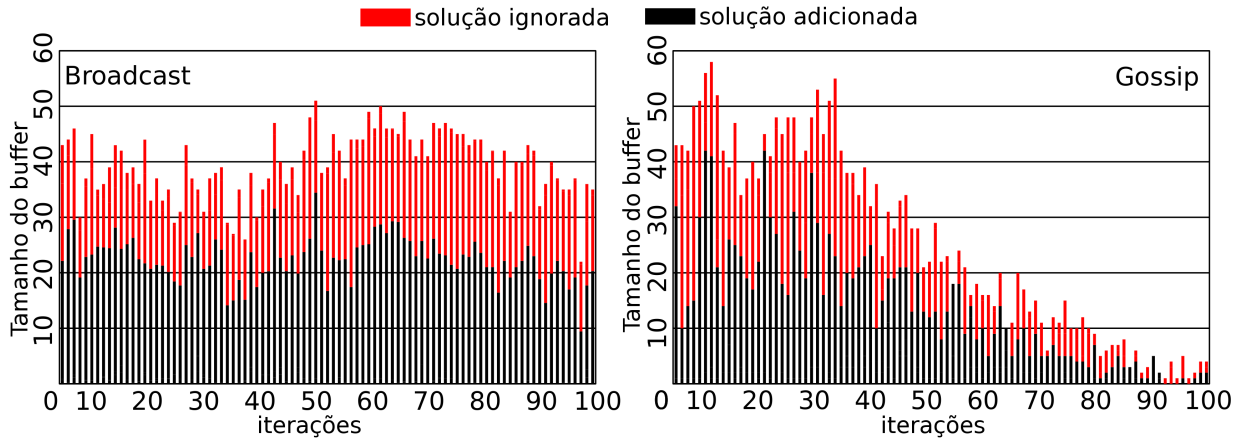


Figura 6.6: Padrão de comunicação para a função DTLZ2 (20 objetivos).

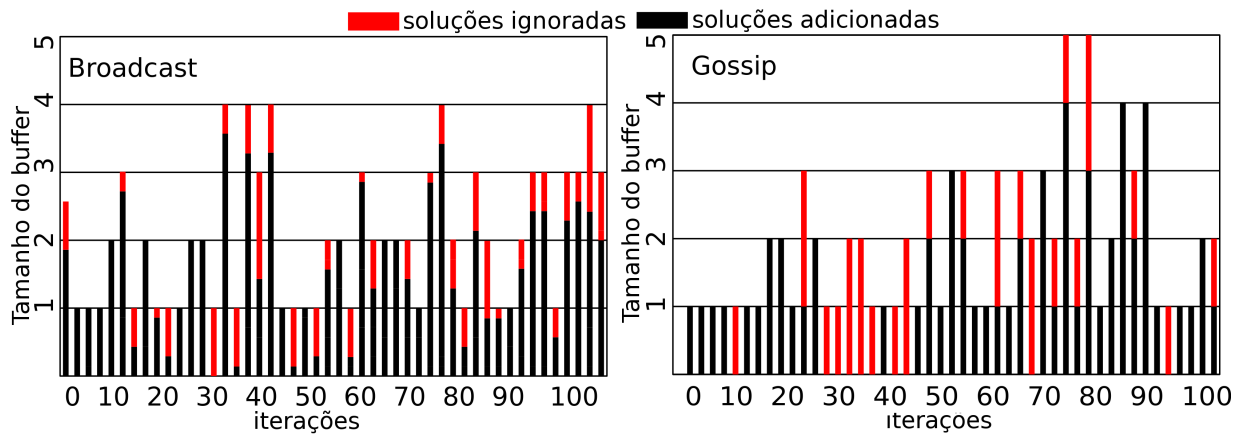


Figura 6.7: Padrão de comunicação para a função DTLZ4 (2 objetivos).

da sobrecarga de comunicação relacionada à estratégia *broadcast*, seu melhor desempenho decorre do fato de que a maioria das soluções compartilhadas entre os enxames foi efetivamente aproveitada.

Para a função DTLZ4 com 20 objetivos, fica claro que a quantidade de soluções não dominadas geradas desde as iterações iniciais é alta: o *buffer* está quase sempre cheio. A Figura 6.8 ilustra o padrão de comunicação de uma execução para as estratégias *broadcast* e *gossip* para resolver a função DTLZ4 com 20 objetivos. A estratégia *gossip* tem a grande maioria das soluções compartilhadas ignoradas pelos enxames receptores. Claramente, a baixa conectividade previne um enxame de se beneficiar das boas soluções obtidas por outros enxames. Assim, quando um enxame envia seu *buffer*, poucas soluções são realmente utilizadas. Por outro lado, o padrão de comunicação *broadcast* mostra que a alta interatividade evita a comunicação de soluções

dominadas, i.e., a comunicação é efetiva.

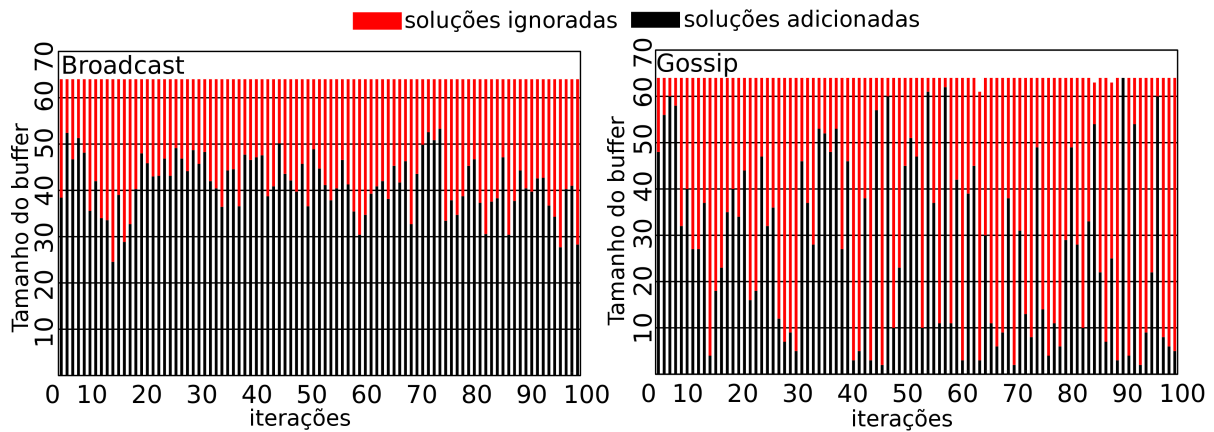


Figura 6.8: Padrão de comunicação para a função DTLZ4 (20 objetivos).

Embora a alta conectividade da estratégia *broadcast* apresente os melhores resultados em vários cenários, para algumas instâncias, a diversidade é afetada e menos comunicação é preferível.

A fim de determinar qual é a melhor abordagem, aplicamos o teste de Friedman considerando os resultados obtidos nas métricas GDp, IGDp e *Spacing* por cada algoritmo. A Figura 6.9 mostra o diagrama *Boxplot* para cada métrica. Para a métrica GDp, de acordo com o teste estatístico, o desempenho da estratégia *broadcast* superou as abordagens centralizada e *gossip*. Para esta métrica, não há diferença estatística entre os resultados obtidos pelos modelos centralizado e *gossip*. Para a métrica IGDp, o teste de Friedman não acusou qualquer diferença estatística entre todos os algoritmos. Porém, os resultados da estratégia *gossip* foram levemente melhores que os resultados das outras duas abordagens, i.e., para esta métrica, uma topologia com baixa quantidade de partículas trocadas pode exercer uma influência positiva nos resultados. Para a métrica *Spacing*, observa-se o mesmo comportamento notado na métrica GDp. Pela observação de todo o conjunto de resultados, podemos concluir que a estratégia de comunicação baseada em *broadcast* apresenta melhor desempenho tanto em termos de convergência quanto de cobertura da fronteira de Pareto.

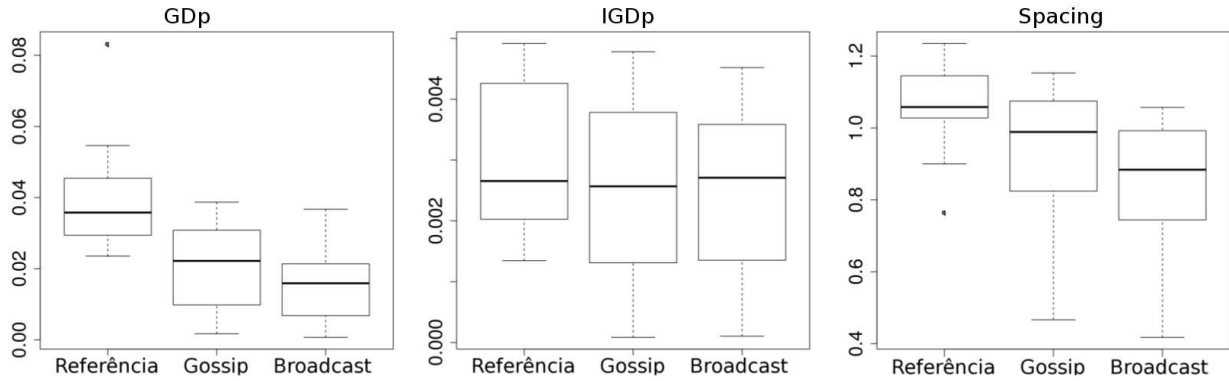


Figura 6.9: Boxplot dos desempenhos médios dos algoritmos para as métricas GDp, IGDp e *Spacing*.

## 6.4 A otimização com múltiplos enxames baseada em decomposição para resolver problemas com vários objetivos

Para avaliar nossos MOPSOs baseados em decomposição com múltiplos enxames, utilizamos o conjunto completo de sete problemas de *benchmarks* da família de testes DTLZ [Deb et al., 2002b]. A análise empírica de nossas propostas considerou experimentos que foram conduzidos em problemas com 2, 3, 5, 10, 15 e 20 objetivos.

Nesta análise, confrontamos o desempenho de MOPSOs com estratégias baseadas em conceitos tradicionais de dominância de Pareto e com estratégias baseadas em decomposição. Para o primeiro grupo, executamos uma versão centralizada e uma versão distribuída do Algoritmo 9 (MOPSO Usando o valor  $\sigma$  com *Crowding Distance*), apresentado no Capítulo 5 e discutido na Seção anterior. Para o segundo grupo, consideramos três algoritmos: uma versão centralizada do algoritmo dMOPSO [Zapotecas Martínez e Coello Coello, 2011], apresentado na Seção 4.4.1 do Capítulo 4 e nossas propostas distribuídas (síncrona e assíncrona) baseadas em decomposição para a resolução de MaOPS.

Em todas as estratégias, as versões centralizadas são versões sequenciais que foram executadas para gerar resultados de referência. Após, os problemas foram executados em duas as versões distribuídas. A otimização termina quando um número pré-definido de 100 iterações é atingido. O mesmo critério de parada foi adotado para as execuções de referência.

A avaliação empírica foi conduzida para investigar o desempenho ao utilizar múltiplos exames para resolver problemas com vários objetivos. Analisamos o comportamento em termos de convergência e diversidade, assim como a escalabilidade em relação ao número de objetivos. Para comparar os resultados, utilizamos um conjunto de indicadores que avaliam a convergência e diversidade dos algoritmos [Zitzler, 1999, Zitzler et al., 2003, Schütze et al., 2012]: GDp, IGDp, *Spacing* e *hypervolume*. Resultados com os menores valores para os indicadores GDp, IGDp e *Spacing* são melhores, enquanto que para o *hypervolume*, valores maiores são melhores.

#### 6.4.1 Simulação: metodologia & parâmetros

Com a intenção de manter a padronização da metodologia adotada, as propostas em questão foram implementadas através da linguagem de programação C utilizando SMPL [MacDougall, 1987] como ferramenta de simulação.

Inicialmente avaliamos os algoritmos sequenciais (Algoritmo 9 e dMOPSO) para gerar resultados referenciais. Para obter esses resultados, os algoritmos utilizaram 256 partículas para executar as DTLZ's 1 e 3 e 128 partículas para executar os problemas remanescentes. Devido ao grande número de problemas, algoritmos e objetivos, adotamos 128 partículas para todos os problemas. Porém, para os problemas DTLZ1 e DTLZ3, observou-se a necessidade um número maior de partículas para melhor representar as fronteiras de Pareto obtidas. Por este motivo, tais problemas foram executados com mais indivíduos. Isto não compromete a investigação proposta, pois o desempenho dos algoritmos é analisado em função de cada problema, individualmente.

Em todas as configurações, cada partícula consistiu de  $n$  valores inicializados aleatoriamente no escopo de cada função. O número total de variáveis é  $n = M + k - 1$ , onde  $M$  é o número de objetivos; 2, 3, 5, 10, 15 e 20 objetivos foram avaliados. Conforme sugerido por Deb *et al.* [Deb et al., 2002b], para a DTLZ1, definimos  $k = 5$ . Para todos os outros problemas,  $k = 10$ . Os valores adotados como parâmetros são tradicionalmente empregados na literatura [Eberhart e Shi, 2000]. O peso de inércia

( $w$ ) foi inicializado em 0,9 e foi gradativamente reduzido para 0,4. As constantes C1 e C2 foram fixadas em 2.

Para o algoritmo baseado na dominância de soluções, o tamanho do repositório foi fixado em função do mesmo número de partículas adotado para cada *benchmark*. Em relação ao algoritmo baseado em decomposição, o número de subproblemas foi fixado em função do mesmo número de partículas adotado para cada *benchmark*.

A mesma abordagem foi empregada para avaliar as versões distribuídas propostas, exceto que o número total de partículas foi distribuído entre 8 enxames: 32 partículas por enxame nas funções DTLZ1 e DTLZ3; 16 partículas por enxame para as demais funções. Seguindo a mesma configuração da versão centralizada, o algoritmo baseado em dominância possui o tamanho do repositório de soluções não dominadas igual ao número de partículas adotado em cada problema. Igualmente, para as versões baseadas em decomposição, o número de problemas decompostos segue a premissa de um subproblema para cada partícula. Esta configuração permite uma comparação justa com os resultados referenciais. A Tabela 6.7 sumariza tais parâmetros. No decorrer desta Seção, os algoritmos abordados foram abreviados da seguinte forma:

- RefPar: algoritmo sequencial de referência, baseado na noção de dominância entre soluções;
- RefDec: algoritmo sequencial de referência, baseado na decomposição do MOP em subproblemas mono-objetivo;
- BPar: algoritmo distribuído, baseado na noção de dominância entre soluções utilizando a topologia *broadcast* entre os enxames;
- BDec: versão síncrona do algoritmo distribuído, baseado na decomposição do MOP em subproblemas mono-objetivo, utilizando a topologia *broadcast* entre os enxames;
- AsBDec: versão assíncrona do algoritmo distribuído, baseado na decomposição

do MOP em subproblemas mono-objetivo, utilizando a topologia *broadcast* entre os enxames.

Tabela 6.7: Configuração adotada para os algoritmos propostos (NA: Não se Aplica).

Algoritmo	DTLZ1 (k=5) e DTLZ3 (k=10)			DTLZ[2,4,5,6,7] (k=10)		
	Partículas	Repositório	Subproblemas	Partículas	Repositório	Subproblemas
<b>RefPar</b>	256	256	NA	128	128	NA
<b>RefDec</b>	256	NA	256	128	NA	128
<b>BPar</b>	32/enxame	32/enxame	NA	16/enxame	16/enxame	NA
<b>BDec</b>	32/enxame	NA	32/enxame	16/enxame	NA	16/enxame
<b>AsBDec</b>	32/enxame	NA	32/enxame	16/enxame	NA	16/enxame

Todos os algoritmos considerados executaram 30 rodadas independentes para cada número de objetivos, e para cada execução, o conteúdo final do repositório foi armazenado. Para as métricas GDp, IGDp e *Spacing*, valores menores são melhores. O oposto aplica-se à métrica *hypervolume*.

A seguir, discutimos os resultados de cada função de *benchmark* e comparamos os resultados obtidos pelos cinco algoritmos. As Tabelas demonstram a média (M.) e o desvio padrão (DP) computado das amostras obtidas das 30 execuções de cada função. Para comparar os resultados, novamente utilizamos o teste de Friedman, o qual foi aplicado com nível de significância de 5% utilizando a ferramenta estatística *R toolkit for statistical computing* [R Development Core Team, 2011]. Resultados destacados em negrito representam valores em que o teste acusou diferença estatística significativa.

Os diagramas *box-and-whisker* e os resultados dos testes estatísticos referentes aos resultados das funções de *benchmarks* estão disponíveis no Apêndice A.

## 6.4.2 Resultados

Os resultados apresentados nas Tabelas 6.8, 6.9, 6.10, 6.11, 6.12, 6.13, 6.14, foram obtidos por todos os algoritmos para os problemas DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6 e DTLZ7, respectivamente. Elas mostram os resultados relativos às métricas GDp, IGDp, *Spacing* e *hypervolume*<sup>3</sup>.

<sup>3</sup>Segundo Yang e Ding [Yang e Ding, 2007], o tempo de execução desta métrica é exponencial em relação a dimensão do espaço de objetivos. Por este motivo, devido a limitações de tempo, para as

Tabela 6.8: Indicadores de desempenho do problema DTLZ1.

DTLZ1 #Obj	Gdp					IGDp				
	RefPar	RefDec	BPar	BDec	AsBDec	RefPar	RefDec	BPar	BDec	AsBDec
2 (M.)	1,3E+2	<b>3,7E+1</b>	4,3E+1	8,1E+1	<b>3,8E+1</b>	1,5E+1	<b>6,9E+0</b>	<b>4,5E+0</b>	1,7E+1	1,1E+1
(DP)	3,6E+1	<b>7,0E+0</b>	4,0E+1	2,4E+1	<b>1,8E+1</b>	4,7E+0	<b>4,3E+0</b>	<b>2,6E+0</b>	8,0E+0	5,9E+0
3 (M.)	1,4E+2	<b>3,4E+1</b>	1,1E+2	<b>3,9E+1</b>	<b>5,0E+1</b>	1,3E+1	<b>5,3E+0</b>	1,4E+1	<b>4,2E+0</b>	8,7E+0
(DP)	2,6E+1	<b>5,7E+0</b>	5,2E+1	<b>1,7E+1</b>	<b>2,8E+1</b>	3,4E+0	<b>3,5E+0</b>	8,7E+0	<b>3,0E+0</b>	6,4E+0
5 (M.)	1,4E+2	<b>4,9E+1</b>	1,5E+2	<b>4,0E+1</b>	<b>3,6E+1</b>	1,2E+1	2,5E+1	2,5E+1	<b>5,9E+0</b>	<b>9,2E+0</b>
(DP)	2,6E+1	<b>7,0E+0</b>	1,4E+1	<b>1,8E+1</b>	<b>1,0E+1</b>	2,6E+0	9,8E+0	1,0E+1	<b>4,2E+0</b>	<b>6,2E+0</b>
10 (M.)	1,5E+2	<b>4,8E+1</b>	1,5E+2	<b>3,5E+1</b>	<b>3,6E+1</b>	<b>1,1E+1</b>	2,1E+1	2,5E+1	<b>5,6E+0</b>	<b>8,3E+0</b>
(DP)	2,9E+1	<b>4,6E+0</b>	1,2E+1	<b>1,6E+1</b>	<b>1,8E+1</b>	<b>4,4E+0</b>	9,0E+0	7,7E+0	<b>3,6E+0</b>	<b>4,7E+0</b>
15 (M.)	1,5E+2	<b>4,8E+1</b>	1,5E+2	<b>3,3E+1</b>	<b>3,1E+1</b>	1,4E+1	2,4E+1	2,8E+1	<b>4,2E+0</b>	<b>9,6E+0</b>
(DP)	3,0E+1	<b>5,4E+0</b>	1,2E+1	<b>1,6E+1</b>	<b>1,3E+1</b>	5,3E+0	7,0E+0	1,0E+1	<b>3,8E+0</b>	<b>6,2E+0</b>
20 (M.)	1,5E+2	<b>4,8E+1</b>	1,5E+2	<b>3,5E+1</b>	<b>3,7E+1</b>	1,5E+1	2,0E+1	2,5E+1	<b>5,0E+0</b>	<b>8,8E+0</b>
(DP)	3,2E+1	<b>5,5E+0</b>	1,1E+1	<b>1,3E+1</b>	<b>1,8E+1</b>	6,5E+0	8,9E+0	9,9E+0	<b>3,2E+0</b>	<b>4,3E+0</b>
DTLZ1 #Obj	Spacing					Hypervolume				
	RefPar	RefDec	BPar	BDec	AsBDec	RefPar	RefDec	BPar	BDec	AsBDec
2 (M.)	1,0E+2	<b>2,0E+1</b>	<b>3,3E+1</b>	5,6E+1	<b>2,2E+1</b>	<b>2,0E+5</b>	<b>2,0E+5</b>	<b>2,0E+5</b>	1,9E+5	1,9E+5
(DP)	2,6E+1	<b>1,2E+0</b>	<b>3,6E+1</b>	1,9E+1	<b>1,4E+1</b>	<b>2,4E+2</b>	<b>1,1E+2</b>	<b>3,0E+2</b>	3,2E+3	2,6E+3
3 (M.)	1,3E+2	<b>1,8E+1</b>	1,0E+2	<b>2,6E+1</b>	<b>3,1E+1</b>	<b>9,0E+7</b>	<b>9,0E+7</b>	<b>8,9E+7</b>	8,9E+7	8,8E+7
(DP)	1,2E+1	<b>9,3E-1</b>	5,5E+1	<b>1,2E+1</b>	<b>2,0E+1</b>	<b>8,0E+3</b>	<b>9,3E+4</b>	<b>9,5E+4</b>	6,5E+5	1,2E+6
5 (M.)	1,3E+2	<b>3,7E+0</b>	1,3E+2	2,5E+1	2,1E+1	<b>8,6E+12</b>	8,1E+12	8,6E+12	8,5E+12	8,4E+12
(DP)	9,1E+0	<b>1,3E+0</b>	1,5E+1	1,4E+1	8,5E+0	<b>1,6E+9</b>	2,0E+11	3,5E+10	9,2E+10	1,3E+11
10 (M.)	1,3E+2	<b>3,9E+0</b>	1,2E+2	2,2E+1	2,1E+1	<b>1,1E+24</b>	1,1E+24	1,1E+24	1,1E+24	1,1E+24
(DP)	1,6E+1	<b>9,0E-1</b>	1,5E+1	1,2E+1	1,2E+1	<b>1,8E+21</b>	2,2E+22	1,1E+22	8,9E+21	1,1E+22
15 (M.)	1,2E+2	<b>4,1E+0</b>	1,1E+2	2,2E+1	1,6E+1	<b>5,4E+30</b>	5,1E+30	5,3E+30	<b>5,3E+30</b>	5,3E+30
(DP)	1,5E+1	<b>6,2E-1</b>	1,6E+1	1,2E+1	9,5E+0	<b>9,4E+27</b>	9,2E+28	5,5E+28	<b>5,3E+28</b>	8,4E+28
20 (M.)	1,2E+2	<b>4,0E+0</b>	1,1E+2	2,2E+1	2,2E+1					
(DP)	1,7E+1	<b>7,1E-1</b>	1,4E+1	1,0E+1	1,3E+1					

O problema DTLZ1 caracteriza-se por possuir um grande número de fronteiras de Pareto locais. Estes subótimos tendem a atrair as soluções antes que elas atinjam a fronteira de Pareto real.

Analisando os resultados é difícil indicar um algoritmo que seja superior aos outros em todos os indicadores para todos os números de objetivos. Porém, algumas constatações são possíveis. Em geral, para este teste, os algoritmos baseados em decomposição apresentam melhor desempenho que as versões baseadas em relações de dominância.

Para 2 e 3 objetivos, o algoritmo RefDec é o único que se destaca com melhores resultados em todos os indicadores. Porém, há de se destacar que em todas as situações, segundo os testes estatísticos, pelo menos uma versão distribuída apresentou desempenho competitivo. Para 5, 10, 15 e 20 objetivos, as versões distribuídas baseadas em decomposição apresentam simultaneamente boas convergência e diversidade, segundo os indicadores Gdp e IGDp. Este entendimento é endossado pela métrica *hypervolume* em relação a versão distribuída síncrona, baseada em de-

DTLZ's[1,2,3,4,6], obtivemos indicadores de *hypervolume* de 2 a 15 objetivos, enquanto que para as DTLZ's[5,7], obtivemos indicadores de *hypervolume* de 2 a 10 objetivos.



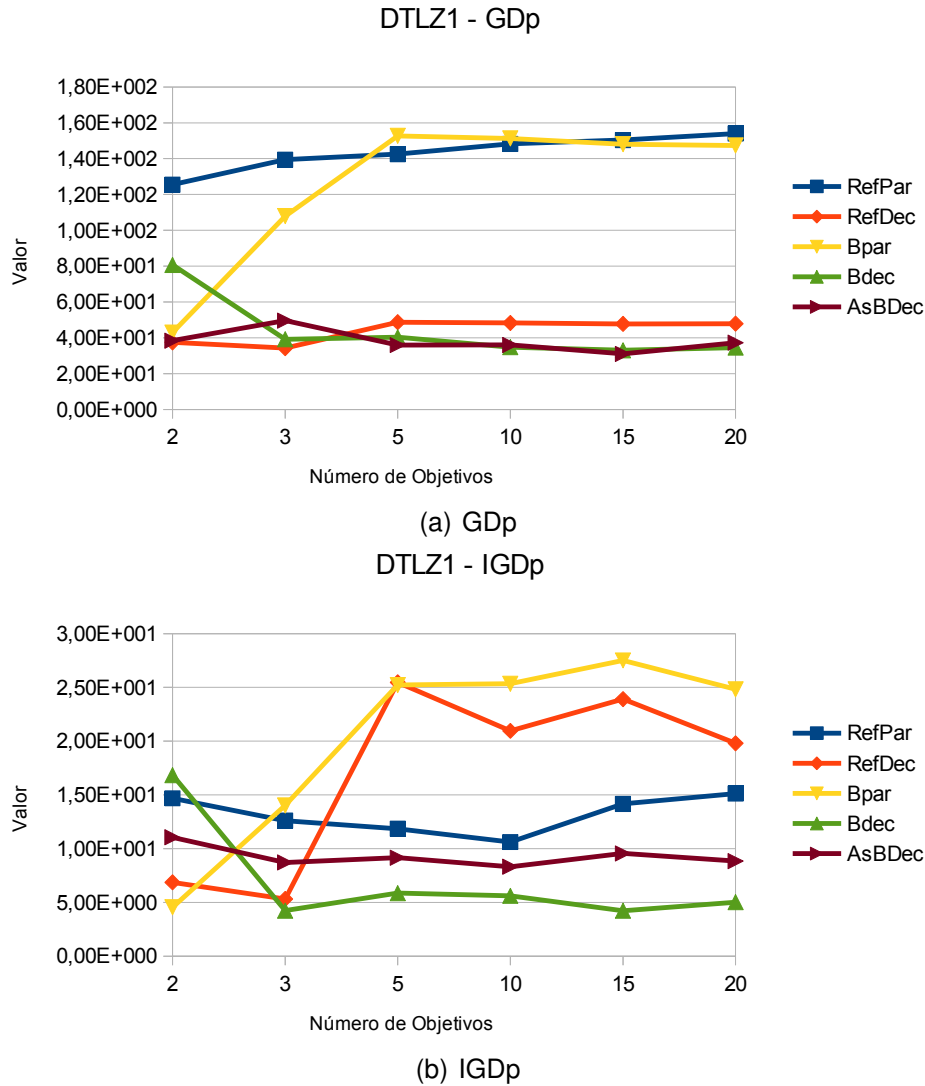


Figura 6.10: Resultados médios relativos aos indicadores GDp e IGDp para o problema DTLZ1.

composição (BDec), para 15 objetivos; embora, para tal métrica, o algoritmo RefPar apresente bons resultados.

A Figura 6.10 ilustra os resultados médios obtidos pelos algoritmos em relação aos indicadores GDp e IGDp. Considerando a escalabilidade dos algoritmos, é possível observar que as versões distribuídas baseadas em decomposição (linhas verde e marrom) são as únicas estratégias que apresentam, simultaneamente, as melhores médias de desempenho para os dois indicadores.

O problema DTLZ2 é um teste adequado para avaliar a convergência dos algoritmos à medida que o número de problemas aumenta. Os indicadores apontam que as versões distribuídas baseadas em decomposição apresentam uma boa escalabili-

Tabela 6.9: Indicadores de desempenho do problema DTLZ2.

DTLZ2 #Obj.	Gdp					IGDp				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	5,5E-1	9,7E-2	2,7E-2	<b>5,3E-3</b>	<b>9,8E-3</b>	1,7E-1	4,1E-2	1,9E-1	<b>1,3E-02</b>	8,0E-2
(DP)	1,7E-1	4,3E-2	1,2E-2	<b>1,4E-2</b>	<b>6,0E-3</b>	4,6E-2	1,1E-2	8,3E-2	<b>1,3E-3</b>	4,9E-2
3 (M.)	6,4E-1	1,8E-1	1,3E-1	<b>1,2E-2</b>	<b>1,6E-1</b>	4,3E-1	3,7E-1	3,7E-1	<b>7,8E-2</b>	<b>1,3E-1</b>
(DP)	1,4E-1	7,7E-2	4,5E-2	<b>5,9E-3</b>	<b>3,5E-1</b>	1,1E-1	1,1E-1	6,8E-2	<b>4,9E-3</b>	<b>3,5E-2</b>
5 (M.)	1,0E+0	2,0E-1	2,8E-1	<b>4,3E-2</b>	<b>5,4E-2</b>	8,5E-1	4,7E-1	5,0E-1	<b>2,5E-1</b>	<b>3,1E-1</b>
(DP)	1,1E-1	5,2E-2	1,2E-1	<b>1,2E-2</b>	<b>7,2E-2</b>	1,6E-1	5,2E-2	9,7E-2	<b>3,8E-2</b>	<b>5,2E-2</b>
10 (M.)	1,2E+0	3,4E-1	4,0E-1	<b>1,7E-2</b>	<b>1,4E-2</b>	9,5E-1	5,3E-1	5,5E-1	<b>4,2E-1</b>	<b>4,8E-1</b>
(DP)	1,0E-1	5,6E-2	1,5E-1	<b>9,6E-3</b>	<b>2,8E-3</b>	1,5E-1	4,0E-2	8,2E-2	<b>4,1E-2</b>	<b>2,9E-2</b>
15 (M.)	1,2E+0	4,4E-1	4,2E-1	<b>1,3E-2</b>	<b>1,2E-2</b>	9,8E-1	5,1E-1	5,3E-1	<b>4,5E-1</b>	<b>5,0E-1</b>
(DP)	1,4E-1	7,0E-2	1,5E-1	<b>3,5E-3</b>	<b>5,9E-3</b>	2,1E-1	3,4E-2	8,6E-2	<b>3,6E-2</b>	<b>7,2E-2</b>
20 (M.)	1,3E+0	4,4E-1	4,5E-1	<b>9,3E-3</b>	<b>1,2E-2</b>	9,0E-1	<b>5,2E-1</b>	<b>5,7E-1</b>	<b>5,2E-1</b>	<b>5,0E-1</b>
(DP)	2,2E-1	1,5E-1	1,6E-1	<b>2,0E-3</b>	<b>3,0E-3</b>	2,2E-1	<b>3,1E-2</b>	<b>9,4E-2</b>	<b>8,3E-2</b>	<b>6,3E-2</b>
DTLZ2 #Obj.	Spacing					Hypervolume				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	9,4E-1	4,9E-1	<b>2,0E-1</b>	5,0E-1	6,0E-1	1,0E+1	<b>1,0E+1</b>	8,7E+0	<b>1,0E+1</b>	1,0E+1
(DP)	1,5E-1	2,5E-2	<b>6,8E-2</b>	1,6E-2	1,7E-2	1,1E-1	<b>2,0E-2</b>	5,8E-1	<b>1,9E-2</b>	4,5E-2
3 (M.)	9,6E-1	<b>3,2E-1</b>	<b>2,9E-1</b>	6,3E-1	7,2E-1	2,9E-1	6,3E-1	7,2E-1	<b>4,7E-2</b>	2,6E-1
(DP)	1,3E-1	<b>7,5E-2</b>	<b>7,7E-2</b>	2,1E-2	1,5E-1	7,7E-2	2,1E-2	1,5E-1	<b>7,0E-3</b>	9,0E-2
5 (M.)	<b>4,7E-2</b>	2,6E-1	4,1E-1	6,1E-1	6,0E-1	<b>5,8E+1</b>	5,7E+1	4,5E+1	<b>5,9E+1</b>	5,8E+1
(DP)	<b>7,0E-3</b>	9,0E-2	1,4E-1	7,7E-2	1,0E-1	<b>4,4E-1</b>	1,3E+0	4,2E+0	<b>3,1E-1</b>	4,8E-1
10 (M.)	1,3E+0	<b>2,4E-1</b>	4,5E-1	4,6E-1	4,8E-1	7,0E+2	6,6E+2	4,9E+2	<b>7,0E+2</b>	6,8E+2
(DP)	1,2E-1	<b>3,4E-2</b>	1,6E-1	1,1E-1	1,7E-1	7,5E+0	4,3E+1	5,0E+1	<b>1,4E+1</b>	2,2E+1
15 (M.)	1,2E+0	<b>2,7E-1</b>	5,0E-1	5,4E-1	5,0E-1	1,8E+5	1,9E+5	1,3E+5	<b>2,0E+5</b>	1,8E+5
(DP)	1,6E-1	<b>3,7E-2</b>	1,4E-1	1,2E-1	2,2E-1	8,5E+3	8,4E+3	1,8E+4	<b>5,1E+3</b>	1,7E+4
20 (M.)	1,2E+0	<b>2,6E-1</b>	4,6E-1	4,9E-1	5,9E-1					
(DP)	1,9E-1	<b>5,3E-2</b>	1,3E-1	1,8E-1	1,6E-1					

dade, com destaque para o algoritmo BDec. Para problemas com qualquer número de objetivos, ele apresentou bons resultados, simultaneamente, nos indicadores GDp, IGDp e *hypervolume*. O fraco desempenho apontado pela métrica *Spacing* deve ser analisado com cautela, pois esta métrica pode apresentar bons resultados de diversidade para fronteiras obtidas distantes da fronteira real.

Um destaque negativo observado neste teste foi o comportamento da versão centralizada baseada em relações de dominância entre soluções (RefPar) que apresentou resultados inferiores em quase todos os indicadores para todos os objetivos.

Ao combinar características dos problemas DTLZ1 e DTLZ2, o *benchmark* DTLZ3 mostra-se adequado para investigar a habilidade dos algoritmos em convergir à fronteira de Pareto real. Para dois objetivos, os algoritmos RefDec e BPar destacam-se em todos os indicadores. Porém, à medida que o número de objetivos aumenta, o algoritmo baseado em decomposição apresenta desempenho superior em termos de convergência e diversidade. Fica evidente que, para este problema, a perceptível superioridade apresentada por este algoritmo é decorrente da decomposição do MOP em subproblemas pertencentes a um único e grande enxame.

Tabela 6.10: Indicadores de desempenho do problema DTLZ3.

DTLZ3 #Obj.	Gdp					IGDp				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	6,0E+2	<b>2,2E+2</b>	<b>3,1E+2</b>	1,4E+3	1,2E+3	2,4E+2	<b>6,0E+1</b>	<b>1,3E+2</b>	3,9E+2	3,8E+2
(DP)	9,6E+1	<b>3,2E+1</b>	<b>9,8E+1</b>	2,2E+2	4,1E+2	4,9E+1	<b>1,8E+1</b>	<b>5,6E+1</b>	1,7E+2	3,4E+2
3 (M.)	7,2E+2	<b>2,1E+2</b>	7,5E+2	1,1E+3	<b>3,4E+2</b>	2,5E+2	<b>6,4E+1</b>	1,9E+2	1,9E+2	1,3E+2
(DP)	1,0E+2	<b>2,3E+1</b>	1,1E+2	2,5E+2	<b>3,6E+2</b>	4,4E+1	<b>2,6E+1</b>	7,8E+1	9,5E+1	7,4E+1
5 (M.)	8,4E+2	<b>2,2E+2</b>	8,8E+2	9,5E+2	7,2E+2	2,9E+2	<b>6,1E+1</b>	3,8E+2	1,9E+2	1,7E+2
(DP)	1,2E+2	<b>2,4E+1</b>	3,0E+1	2,5E+2	4,2E+2	4,4E+1	<b>2,6E+1</b>	7,1E+1	1,1E+2	9,7E+1
10 (M.)	9,3E+2	<b>2,2E+2</b>	9,1E+2	9,4E+2	9,1E+2	3,6E+2	<b>5,2E+1</b>	3,9E+2	<b>1,5E+2</b>	2,1E+2
(DP)	1,2E+2	<b>1,5E+0</b>	2,9E+1	2,8E+2	3,2E+2	6,1E+1	<b>2,8E+1</b>	7,2E+1	<b>1,1E+2</b>	8,6E+1
15 (M.)	9,4E+2	<b>2,2E+2</b>	9,1E+2	1,0E+3	1,1E+3	3,3E+2	<b>5,8E+1</b>	4,0E+2	2,0E+2	2,5E+2
(DP)	1,2E+2	<b>1,8E+0</b>	3,8E+1	2,3E+2	3,1E+2	5,7E+1	<b>2,7E+1</b>	9,2E+1	1,2E+2	8,1E+1
20 (M.)	9,4E+2	<b>2,2E+2</b>	9,2E+2	9,5E+2	9,3E+2	3,4E+2	<b>6,0E+1</b>	4,0E+2	2,0E+2	2,4E+2
(DP)	1,1E+2	<b>1,8E+0</b>	3,8E+1	2,1E+2	3,4E+2	5,8E+1	<b>2,6E+1</b>	8,1E+1	1,1E+2	1,1E+2
DTLZ3 #Obj.	Spacing					Hypervolume				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	4,5E+2	<b>1,2E+2</b>	<b>2,3E+2</b>	8,9E+2	7,0E+2	3,9E+6	<b>3,9E+6</b>	<b>3,9E+6</b>	3,2E+6	3,3E+6
(DP)	6,7E+1	<b>9,2E+0</b>	<b>8,9E+1</b>	1,3E+2	2,1E+2	2,1E+4	<b>2,3E+3</b>	<b>1,9E+4</b>	3,5E+5	6,6E+5
3 (M.)	6,4E+2	<b>1,3E+2</b>	6,8E+2	6,7E+2	<b>1,7E+2</b>	5,7E+9	<b>5,7E+9</b>	5,7E+9	5,2E+9	5,5E+9
(DP)	5,0E+1	<b>1,1E+1</b>	9,4E+1	1,5E+2	<b>2,1E+2</b>	6,5E+6	<b>1,7E+6</b>	2,2E+7	2,6E+8	2,6E+8
5 (M.)	6,8E+2	<b>5,6E+1</b>	6,8E+2	5,9E+2	4,3E+2	<b>1,5E+16</b>	<b>1,5E+16</b>	1,4E+16	1,3E+16	1,4E+16
(DP)	2,0E+1	<b>1,0E+1</b>	4,2E+1	1,5E+2	2,6E+2	<b>6,9E+13</b>	<b>1,6E+13</b>	2,3E+14	7,8E+14	6,2E+14
10 (M.)	6,7E+2	<b>3,8E+1</b>	6,3E+2	5,8E+2	5,3E+2	5,6E+31	<b>5,8E+31</b>	5,3E+31	5,3E+31	5,3E+31
(DP)	3,0E+1	<b>3,0E+0</b>	4,4E+1	1,9E+2	2,0E+2	4,8E+29	<b>1,8E+28</b>	2,0E+30	3,0E+30	2,7E+30
15 (M.)	6,3E+2	<b>4,1E+1</b>	6,0E+2	6,5E+2	6,0E+2	8,2E+44	<b>8,6E+44</b>	7,6E+44	7,8E+44	7,6E+44
(DP)	7,2E+1	<b>3,4E+0</b>	7,5E+1	1,5E+2	1,8E+2	1,4E+43	<b>6,3E+40</b>	3,6E+43	4,4E+43	3,3E+43
20 (M.)	6,0E+2	<b>4,0E+1</b>	5,7E+2	5,9E+2	5,1E+2					
(DP)	5,0E+1	<b>2,7E+0</b>	6,2E+1	1,3E+2	1,7E+2					

Através da Figura 6.11 é possível observar que o algoritmo RefDec apresenta boa escalabilidade. E que os bons resultados de convergência apontados pela métrica GDp são acompanhados de soluções bem distribuídas ao longo das fronteiras, conforme indicado pela métrica *Spacing*.

Em relação ao problema DTLZ4, ele tende a gerar pontos em uma região específica na fronteira de Pareto. Por este motivo, ele é indicado para avaliar a distribuição das soluções. Para dois objetivos, o algoritmo distribuído e síncrono (BDec) baseado em decomposição apresenta bom desempenho em todos os indicadores. Desempenho similar é observado pelo algoritmo centralizado baseado em decomposição, embora a métrica *Spacing* aponte que suas soluções não apresentam boa diversidade.

À medida que o número de objetivos aumenta, a superioridade isolada do algoritmo BDec deixa de existir. Todas as versões baseadas em decomposição mostram-se competitivas. Todavia, é possível notar que as versões distribuídas baseadas em decomposição apresentam os melhores resultados em termos de convergência e distribuição (pelos indicadores GDp e *Spacing*, respectivamente). O bom desempenho

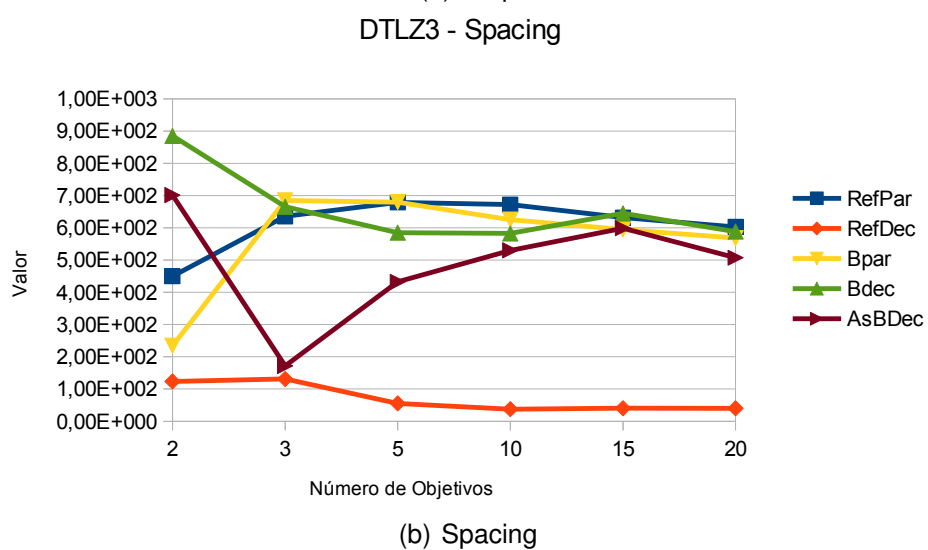
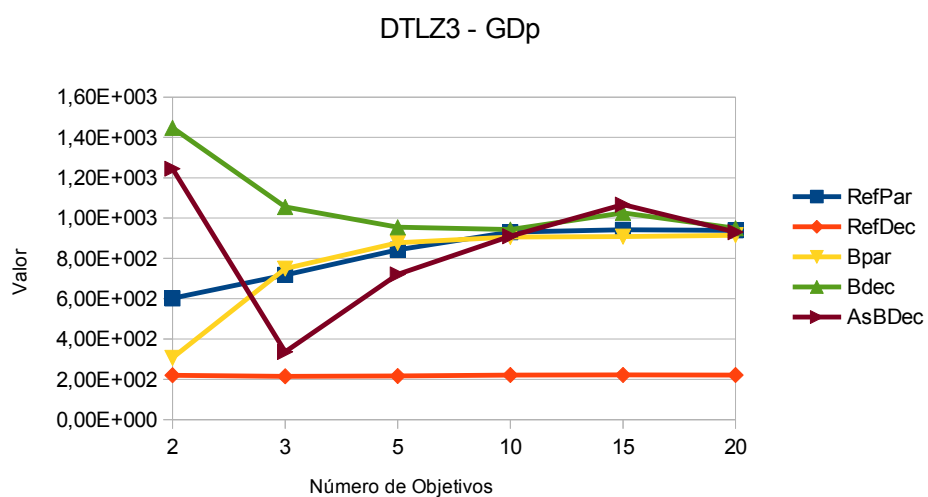


Figura 6.11: Resultados médios relativos aos indicadores GDp e Spacing para o problema DTLZ3.

Tabela 6.11: Indicadores de desempenho do problema DTLZ4.

DTLZ4 #Obj.	Gdp					IGDp				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	2,7E-1	<b>1,3E-1</b>	4,2E-1	<b>7,8E-2</b>	<b>1,9E-1</b>	9,9E-2	<b>5,5E-2</b>	<b>6,1E-2</b>	<b>5,5E-2</b>	1,6E-1
(DP)	1,1E-1	<b>3,6E-2</b>	2,0E-1	<b>2,3E-1</b>	<b>2,8E-1</b>	2,4E-2	<b>1,5E-2</b>	<b>2,4E-2</b>	<b>9,5E-2</b>	1,0E-1
3 (M.)	4,6E-1	5,9E-2	6,9E-2	<b>7,6E-3</b>	<b>7,4E-2</b>	<b>1,2E-1</b>	<b>1,5E-1</b>	<b>1,2E-1</b>	2,1E-1	2,2E-1
(DP)	1,1E-1	5,0E-2	3,5E-2	<b>2,4E-2</b>	<b>1,3E-1</b>	<b>2,0E-2</b>	<b>2,9E-2</b>	<b>5,1E-2</b>	6,2E-2	7,0E-2
5 (M.)	6,8E-1	5,8E-2	1,8E-1	<b>2,4E-3</b>	<b>1,2E-3</b>	2,9E-1	<b>2,2E-1</b>	3,1E-1	2,9E-1	2,9E-1
(DP)	1,2E-1	5,4E-2	6,3E-2	<b>2,7E-3</b>	<b>1,8E-3</b>	4,5E-2	<b>2,3E-2</b>	2,2E-2	1,5E-2	1,7E-2
10 (M.)	7,3E-1	1,2E-1	3,6E-1	<b>2,3E-2</b>	<b>4,8E-2</b>	3,7E-1	<b>2,8E-1</b>	3,7E-1	3,2E-1	3,6E-1
(DP)	9,9E-2	7,4E-2	1,1E-1	<b>1,8E-2</b>	<b>1,6E-1</b>	4,5E-2	<b>2,3E-2</b>	2,8E-2	2,1E-2	1,8E-1
15 (M.)	7,4E-1	<b>1,5E-1</b>	4,1E-1	<b>6,9E-2</b>	<b>6,1E-2</b>	4,2E-1	<b>3,5E-1</b>	4,2E-1	5,6E-1	5,2E-1
(DP)	1,0E-1	<b>7,3E-2</b>	1,1E-1	<b>5,3E-2</b>	<b>7,2E-2</b>	5,0E-2	<b>1,4E-2</b>	2,6E-2	3,7E-1	3,4E-1
20 (M.)	7,5E-1	<b>1,3E-1</b>	4,3E-1	<b>7,3E-2</b>	<b>1,0E-1</b>	4,9E-1	<b>4,3E-1</b>	5,1E-1	6,8E-1	6,2E-1
(DP)	1,2E-1	<b>1,0E-1</b>	1,0E-1	<b>5,9E-2</b>	<b>1,3E-1</b>	5,3E-2	<b>1,8E-2</b>	3,7E-2	3,7E-1	3,4E-1
DTLZ4 #Obj.	Spacing					Hypervolume				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	7,4E-1	6,2E-1	5,8E-1	<b>4,3E-1</b>	<b>3,3E-1</b>	6,8E+0	<b>7,0E+0</b>	6,8E+0	<b>6,8E+0</b>	6,3E+0
(DP)	1,2E-1	9,7E-2	1,4E-1	<b>2,4E-1</b>	<b>2,8E-1</b>	1,4E-1	<b>4,4E-2</b>	2,1E-1	<b>5,3E-1</b>	5,9E-1
3 (M.)	1,0E+0	<b>1,5E-1</b>	<b>2,4E-1</b>	<b>3,2E-1</b>	<b>2,2E-1</b>	<b>2,2E+1</b>	<b>2,2E+1</b>	2,0E+1	2,0E+1	1,9E+1
(DP)	1,3E-1	<b>8,5E-2</b>	<b>1,1E-1</b>	<b>2,5E-1</b>	<b>2,5E-1</b>	<b>3,0E-1</b>	<b>4,0E-1</b>	1,4E+0	2,2E+0	2,0E+0
5 (M.)	5,6E-1	1,3E-1	7,8E-2	<b>4,8E-2</b>	<b>2,8E-2</b>	<b>9,3E+1</b>	<b>1,1E+2</b>	7,0E+1	7,8E+1	7,4E+1
(DP)	1,2E-1	2,9E-2	4,0E-2	<b>1,2E-1</b>	<b>1,1E-1</b>	<b>6,5E+0</b>	<b>4,4E+0</b>	6,9E+0	1,2E+1	7,8E+0
10 (M.)	5,9E-1	2,0E-1	1,9E-1	<b>1,2E-1</b>	<b>8,2E-2</b>	2,0E+4	<b>2,3E+4</b>	1,7E+4	1,9E+4	1,8E+4
(DP)	1,7E-1	6,0E-2	9,4E-2	<b>2,2E-1</b>	<b>1,7E-1</b>	1,4E+3	<b>5,7E+2</b>	1,2E+3	2,2E+3	1,9E+3
15 (M.)	6,8E-1	2,0E-1	2,5E-1	<b>1,7E-1</b>	<b>7,3E-2</b>	1,9E+6	<b>2,1E+6</b>	1,4E+6	1,6E+6	1,5E+6
(DP)	1,7E-1	7,3E-2	1,1E-1	<b>2,1E-1</b>	<b>1,1E-1</b>	1,5E+5	<b>5,7E+4</b>	1,8E+5	2,7E+5	2,6E+5
20 (M.)	7,2E-1	<b>1,8E-1</b>	<b>2,1E-1</b>	<b>2,3E-1</b>	<b>1,0E-1</b>					
(DP)	1,9E-1	<b>1,0E-1</b>	<b>7,7E-2</b>	<b>2,9E-1</b>	<b>1,4E-1</b>					

destas versões distribuídas é ilustrado pela Figura 6.12.

Considerando as versões baseadas na dominância entre as soluções, a métrica *hypervolume* aponta que para dois objetivos, a versão distribuída (BPar) é superior à versão centralizada (RefPar). Entretanto, à medida que o número de objetivos aumenta, esta relação se inverte.

Em relação ao teste DTLZ5, a fronteira de Pareto real é apenas uma parte da unidade esférica e não uma área, i.e., a fronteira é uma curva densamente populada em um espaço de objetivos M-dimensional. Considerando o indicador GDp, fica evidente a superioridade do algoritmo distribuído baseado em relações de dominância de Pareto (Bpar). Os testes estatísticos o apontam como o melhor algoritmo para qualquer número de objetivos. Similar constatação pode ser deduzida em relação à métrica *Spacing*, embora o algoritmo RefPar apresente bons resultados em algumas situações.

Considerando o indicador IGDp, destacam-se os desempenhos das versões distribuídas baseadas em decomposição, embora o algoritmo distribuído BPar também apresente bons resultados pra 2, 15 e 20 objetivos. A Figura 6.13(a) destaca o bom

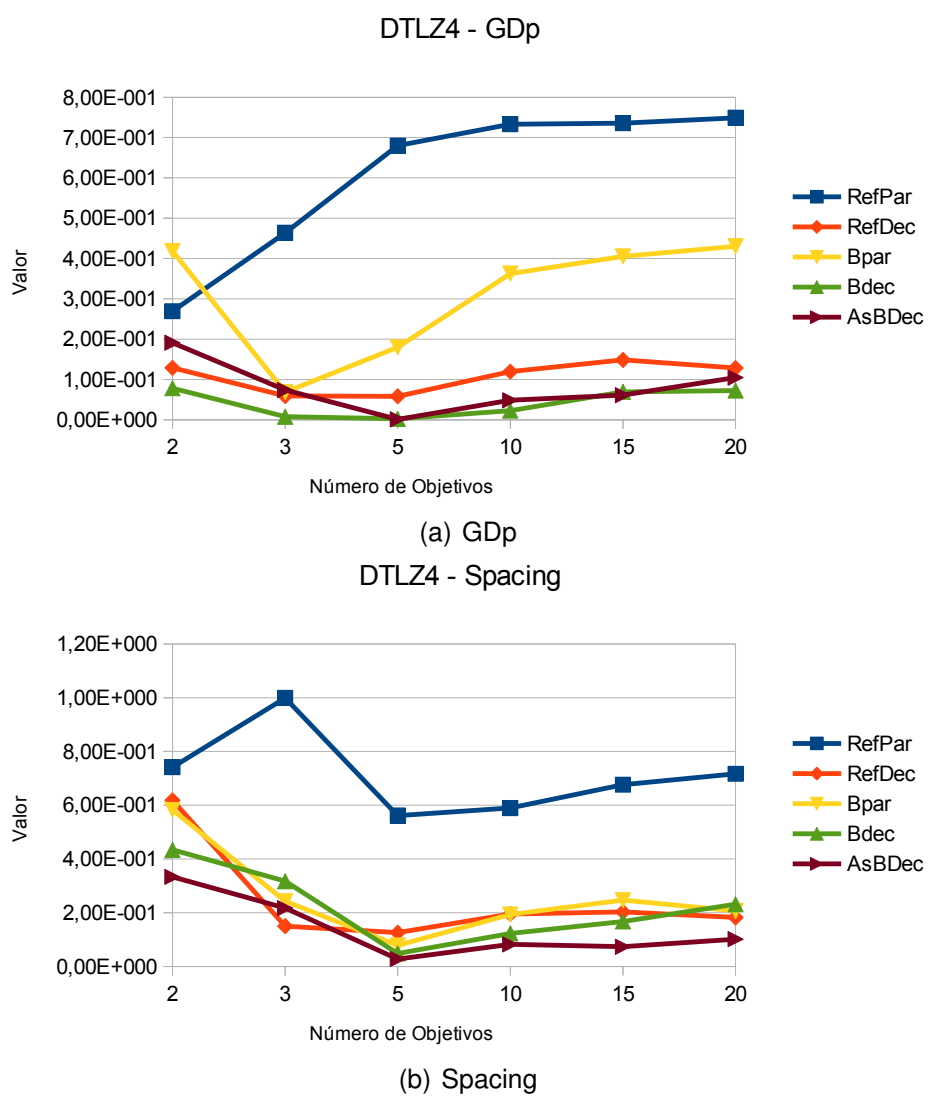


Figura 6.12: Resultados médios relativos aos indicadores GDp e *Spacing* para o problema DTLZ4.

Tabela 6.12: Indicadores de desempenho do problema DTLZ5.

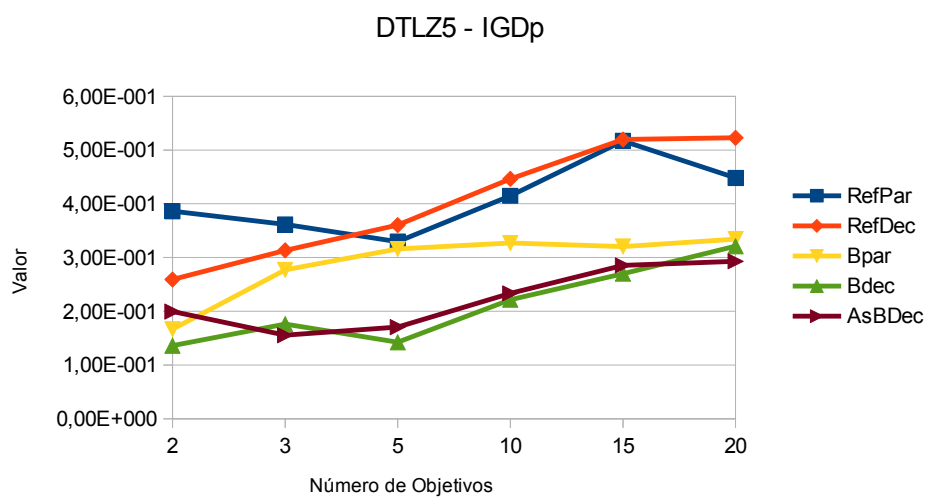
DTLZ5 #Obj.	Gdp					IGDp				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	5,8E-1	1,3E+0	<b>2,0E-1</b>	4,9E-1	5,2E-1	3,9E-1	2,6E-1	<b>1,7E-1</b>	<b>1,4E-1</b>	<b>2,0E-1</b>
(DP)	1,5E-1	2,1E-1	<b>9,5E-2</b>	4,7E-2	8,6E-2	1,0E-1	3,4E-2	<b>5,9E-2</b>	<b>1,8E-2</b>	<b>4,0E-2</b>
3 (M.)	6,9E-1	1,3E+0	<b>2,9E-1</b>	6,2E-1	4,7E-1	3,6E-1	3,1E-1	2,8E-1	1,8E-1	<b>1,6E-1</b>
(DP)	1,1E-1	2,0E-1	<b>1,0E-1</b>	5,8E-2	<b>6,6E-2</b>	4,7E-2	5,5E-2	9,1E-2	<b>2,2E-2</b>	<b>2,6E-2</b>
5 (M.)	6,1E-1	1,2E+0	<b>3,1E-1</b>	6,3E-1	5,7E-1	3,3E-1	3,6E-1	3,2E-1	<b>1,4E-1</b>	<b>1,7E-1</b>
(DP)	1,1E-1	1,8E-1	<b>1,0E-1</b>	1,3E-1	2,3E-1	7,7E-2	6,3E-2	1,0E-1	<b>2,8E-2</b>	<b>3,8E-2</b>
10 (M.)	5,9E-1	1,2E+0	<b>3,2E-1</b>	5,7E-1	6,2E-1	4,1E-1	4,5E-1	3,3E-1	<b>2,2E-1</b>	<b>2,3E-1</b>
(DP)	1,6E-1	1,5E-1	<b>1,0E-1</b>	1,3E-1	2,3E-1	1,4E-1	5,7E-2	1,0E-1	<b>5,4E-2</b>	<b>8,5E-2</b>
15 (M.)	6,3E-1	1,3E+0	<b>3,0E-1</b>	9,0E-1	6,7E-1	5,2E-1	5,2E-1	<b>3,2E-1</b>	<b>2,7E-1</b>	<b>2,9E-1</b>
(DP)	1,2E-1	1,5E-1	<b>1,4E-1</b>	4,1E-1	2,6E-1	1,3E-1	7,4E-2	<b>1,1E-1</b>	<b>5,9E-2</b>	<b>5,8E-2</b>
20 (M.)	5,3E-1	1,4E+0	<b>2,8E-1</b>	5,1E-1	6,5E-1	4,5E-1	5,2E-1	<b>3,3E-1</b>	<b>3,2E-1</b>	<b>2,9E-1</b>
(DP)	1,8E-1	3,3E-1	<b>8,7E-2</b>	1,1E-1	1,5E-1	1,4E-1	8,8E-2	<b>9,9E-2</b>	<b>7,8E-2</b>	<b>6,1E-2</b>
DTLZ5 #Obj.	Spacing					Hypervolume				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	8,4E-1	1,3E+0	<b>4,6E-1</b>	7,8E-1	9,6E-1	1,4E+1	1,5E+1	1,4E+1	<b>1,5E+1</b>	<b>1,5E+1</b>
(DP)	9,7E-2	5,6E-2	<b>1,1E-1</b>	4,2E-2	7,3E-2	2,9E-1	1,0E-1	7,1E-1	<b>8,7E-2</b>	<b>7,7E-2</b>
3 (M.)	7,5E-1	1,2E+0	<b>3,9E-1</b>	7,9E-1	8,0E-1	3,4E+1	<b>3,6E+1</b>	3,2E+1	<b>3,6E+1</b>	3,5E+1
(DP)	5,7E-2	5,3E-2	<b>1,5E-1</b>	5,0E-2	5,5E-2	5,9E-1	<b>4,4E-1</b>	3,8E+0	<b>4,9E-1</b>	7,3E-1
5 (M.)	<b>5,3E-1</b>	9,0E-1	<b>3,4E-1</b>	8,6E-1	8,9E-1	2,2E+2	2,4E+2	1,9E+2	<b>2,5E+2</b>	2,4E+2
(DP)	<b>7,7E-2</b>	4,7E-2	<b>1,3E-1</b>	1,1E-1	1,8E-1	2,3E+1	5,5E+0	3,2E+1	<b>4,1E+0</b>	8,0E+0
10 (M.)	<b>4,4E-1</b>	6,8E-1	<b>3,4E-1</b>	8,7E-1	9,2E-1	6,7E+4	7,2E+4	5,4E+4	<b>7,7E+4</b>	<b>7,7E+4</b>
(DP)	<b>1,4E-1</b>	2,8E-2	<b>1,4E-1</b>	1,3E-1	2,7E-1	1,1E+4	2,6E+3	1,2E+4	<b>1,3E+3</b>	<b>2,2E+3</b>
15 (M.)	<b>4,1E-1</b>	7,3E-1	<b>3,4E-1</b>	1,2E+0	1,0E+0					
(DP)	<b>1,6E-1</b>	2,5E-2	<b>1,3E-1</b>	3,7E-1	2,1E-1					
20 (M.)	<b>4,1E-1</b>	8,0E-1	<b>3,1E-1</b>	7,2E-1	1,1E+0					
(DP)	<b>1,6E-1</b>	9,5E-2	<b>1,3E-1</b>	1,8E-1	1,2E-1					

desempenho obtido pelas versões distribuídas, à medida que o número de objetivos aumenta.

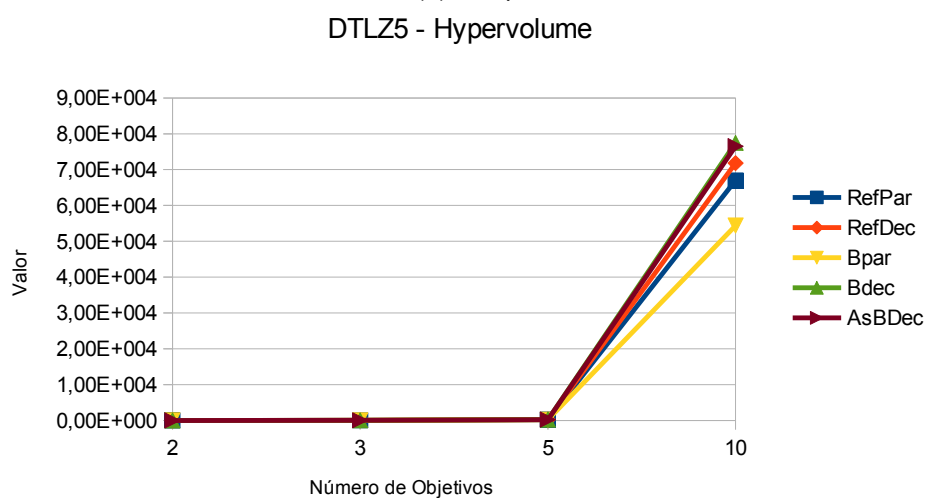
Em relação à convergência dos resultados, neste problema é perceptível a superioridade dos algoritmos distribuídos. A única situação em que um algoritmo centralizado apresenta desempenho similar às propostas distribuídas é observada para 3 objetivos (apenas na métrica *hypervolume*). Ainda considerando a métrica *hypervolume*, é possível observar através da Figura 6.13(b) o bom desempenho dos algoritmos distribuídos baseados em decomposição, quando vários objetivos são considerados.

O *benchmark* DTLZ6 é uma variação do problema anterior e prioriza avaliar a convergência dos algoritmos. Considerando os indicadores GDp, IGDp e *Spacing*, fica evidente a superioridade dos algoritmos baseados em decomposição, com destaque para a versão sequencial.

Neste problema, podemos constatar que decompor o MOP em subproblemas pertencentes a um único e grande enxame mostra-se uma boa alternativa. Em relação à métrica *hypervolume*, fica difícil assinalar um algoritmo que se destaque em relação aos outros, principalmente à medida que o número de objetivos aumenta. O teste es-



(a) IGDp



(b) Hypervolume

Figura 6.13: Resultados médios relativos aos indicadores IGDp e Hypervolume para o problema DTLZ5.



Tabela 6.13: Indicadores de desempenho do problema DTLZ6.

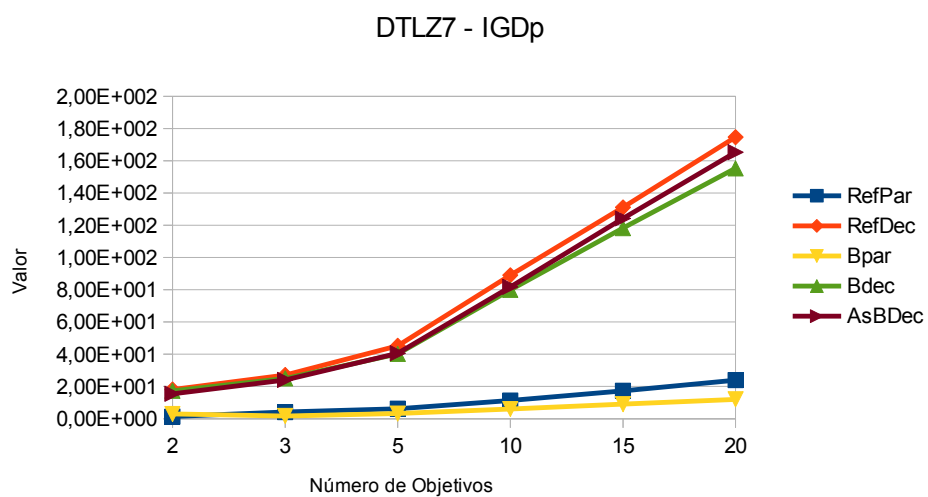
DTLZ6 #Obj.	Gdp					IGDp				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	8,4E+0	<b>6,9E-4</b>	7,9E+0	8,0E+0	<b>8,7E-1</b>	7,9E+0	<b>9,1E-3</b>	8,6E-1	7,8E+0	1,1E+0
(DP)	1,2E-1	<b>3,2E-3</b>	3,1E-1	6,7E-2	<b>9,7E-1</b>	1,3E-1	<b>2,2E-3</b>	0,0E+0	9,3E-2	4,3E-1
3 (M.)	8,9E+0	<b>5,7E-3</b>	9,1E+0	8,1E+0	<b>2,2E+0</b>	8,5E+0	<b>2,2E-1</b>	4,6E+0	7,8E+0	2,5E+0
(DP)	1,3E-1	<b>1,7E-2</b>	1,3E-1	1,1E-1	<b>1,5E+0</b>	1,5E-1	<b>2,7E-1</b>	4,1E+0	1,4E-1	1,2E+0
5 (M.)	9,3E+0	<b>2,5E-2</b>	9,2E+0	7,9E+0	7,8E+0	8,8E+0	<b>7,3E-1</b>	8,9E+0	7,9E+0	7,9E+0
(DP)	5,8E-2	<b>1,4E-1</b>	1,0E-1	1,7E-1	1,6E-1	1,5E-1	<b>3,4E-2</b>	2,1E-1	1,3E-1	1,4E-1
10 (M.)	9,3E+0	<b>2,8E-2</b>	9,3E+0	7,9E+0	7,9E+0	8,9E+0	<b>7,8E-1</b>	9,0E+0	8,1E+0	8,1E+0
(DP)	7,7E-2	<b>1,5E-1</b>	1,2E-1	2,0E-1	1,7E-1	1,5E-1	<b>5,5E-2</b>	2,1E-1	2,1E-1	1,7E-1
15 (M.)	9,2E+0	<b>2,7E-2</b>	9,2E+0	7,9E+0	7,9E+0	8,8E+0	<b>8,2E-1</b>	8,9E+0	8,2E+0	8,1E+0
(DP)	9,6E-2	<b>1,5E-1</b>	8,8E-2	1,9E-1	1,9E-1	1,3E-1	<b>1,1E-2</b>	2,1E-1	2,0E-1	2,1E-1
20 (M.)	9,2E+0	<b>2,7E-2</b>	9,3E+0	7,8E+0	8,0E+0	8,8E+0	<b>8,2E-1</b>	8,9E+0	8,2E+0	8,2E+0
(DP)	1,1E-1	<b>1,5E-1</b>	8,2E-2	2,5E-1	1,7E-1	1,5E-1	<b>8,5E-3</b>	1,8E-1	2,5E-1	1,9E-1
DTLZ6 #Obj.	Spacing					Hypervolume				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	4,9E+0	<b>4,9E-1</b>	6,2E+0	4,4E+0	<b>5,9E-1</b>	5,0E+1	<b>1,2E+2</b>	<b>1,1E+2</b>	4,9E+1	1,1E+2
(DP)	3,1E-1	<b>3,0E-2</b>	5,1E-1	1,8E-1	<b>6,6E-1</b>	1,1E+0	<b>1,1E-3</b>	<b>2,9E-1</b>	7,8E-1	6,2E+0
3 (M.)	4,4E+0	<b>2,5E-1</b>	3,2E+0	5,5E+0	<b>4,0E-1</b>	4,3E+2	<b>1,1E+3</b>	7,0E+2	5,9E+2	<b>8,5E+2</b>
(DP)	3,3E-1	<b>1,2E-1</b>	1,1E+0	3,0E-1	<b>5,7E-1</b>	2,0E+1	<b>4,3E+1</b>	3,8E+2	1,8E+1	<b>1,3E+2</b>
5 (M.)	5,2E+0	<b>4,7E-2</b>	3,2E+0	2,8E+0	1,7E+0	<b>5,5E+4</b>	<b>1,1E+5</b>	2,8E+4	4,4E+4	3,3E+4
(DP)	2,3E-1	<b>7,0E-3</b>	8,7E-1	9,6E-1	9,9E-1	<b>1,6E+3</b>	<b>1,0E+2</b>	6,4E+3	5,6E+3	7,0E+3
10 (M.)	5,3E+0	<b>2,5E-2</b>	3,0E+0	7,4E-1	<b>2,9E-1</b>	<b>1,5E+9</b>	<b>3,1E+9</b>	<b>6,3E+8</b>	<b>7,4E+8</b>	<b>5,4E+8</b>
(DP)	4,1E-1	<b>2,3E-2</b>	1,3E+0	4,7E-1	<b>3,0E-1</b>	<b>4,9E+7</b>	<b>4,6E+6</b>	<b>1,9E+8</b>	<b>1,2E+8</b>	<b>1,0E+8</b>
15 (M.)	4,4E+0	<b>1,0E-2</b>	3,1E+0	<b>6,1E-1</b>	<b>2,5E-1</b>	<b>1,2E+11</b>	<b>2,7E+11</b>	<b>5,2E+10</b>	<b>5,8E+10</b>	<b>4,2E+10</b>
(DP)	5,5E-1	<b>3,2E-3</b>	9,7E-1	<b>3,1E-1</b>	<b>2,4E-1</b>	<b>6,9E+9</b>	<b>2,0E+7</b>	<b>1,1E+10</b>	<b>8,4E+9</b>	<b>7,3E+9</b>
20 (M.)	4,5E+0	<b>6,1E-3</b>	3,3E+0	<b>5,0E-2</b>	<b>1,9E-1</b>					
(DP)	6,5E-1	<b>2,0E-3</b>	1,0E+0	<b>1,2E-1</b>	<b>1,8E-1</b>					

estatístico apontou que para esta métrica, todos os problemas apresentam desempenho equivalente em 10 e 15 objetivos.

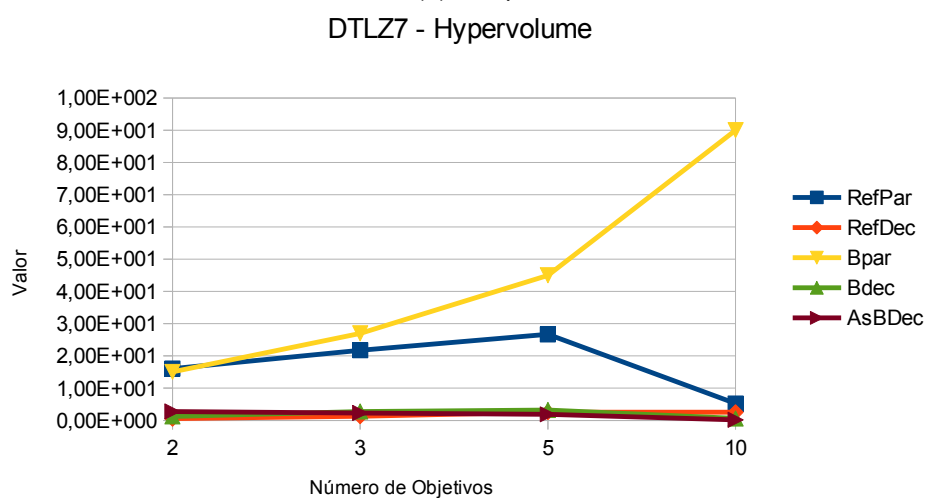
Finalmente, a DTLZ7 testa a habilidade dos algoritmos em manter indivíduos em diferentes regiões da fronteira de Pareto real. Neste problema, destacam-se os algoritmos baseados em relações de dominância entre as soluções. Tanto a versão distribuída (BPar) quanto a versão centralizada (RefPar) apresentam os melhores índices para todas as métricas.

O desempenho dos algoritmos avaliados em relação ao problema DTLZ7 pode ser visualizado pela Figura 6.14. A Figura 6.14(a) ilustra os resultados obtidos pelo indicador IGDp, enquanto que a Figura 6.14(b) ilustra os resultados referentes à métrica *hypervolume*. Em ambas as situações, é perceptível o superior desempenho do algoritmo distribuído BPar, principalmente em relação aos valores de *hypervolume* obtidos à medida que o número de objetivos aumenta.

Para este problema, fica evidente que decompor um MOP em subproblemas em um único e centralizado enxame é uma péssima estratégia. Tal constatação reforça a necessidade de interpretar com ressalvas os resultados da métrica *Spacing*. Este indi-



(a) IGDp



(b) Hypervolume

Figura 6.14: Resultados médios relativos aos indicadores IGDp e Hypervolume para o problema DTLZ7.

Tabela 6.14: Indicadores de desempenho do problema DTLZ7.

DTLZ7 #Obj.	Gdp					IGDp				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	<b>3,7E+0</b>	1,8E+1	<b>5,1E+0</b>	1,7E+1	1,6E+1	<b>1,4E+0</b>	1,8E+1	<b>3,0E+0</b>	1,7E+1	1,6E+1
(DP)	<b>9,0E-1</b>	6,4E-2	<b>5,7E-1</b>	2,8E-1	4,3E-1	<b>4,4E-1</b>	4,6E-1	<b>1,0E+0</b>	4,6E-1	8,9E-1
3 (M.)	<b>8,7E+0</b>	2,7E+1	9,0E+0	2,5E+1	2,4E+1	<b>4,2E+0</b>	2,7E+1	<b>1,7E+0</b>	2,5E+1	2,4E+1
(DP)	<b>9,7E-1</b>	7,6E-2	9,7E-1	4,9E-1	4,7E-1	<b>5,7E-1</b>	5,9E-1	<b>1,9E-1</b>	1,1E+0	1,2E+0
5 (M.)	<b>7,0E+0</b>	4,5E+1	<b>1,8E+1</b>	4,1E+1	4,0E+1	<b>6,2E+0</b>	4,5E+1	<b>3,2E+0</b>	4,0E+1	4,1E+1
(DP)	<b>9,1E-1</b>	4,5E-1	<b>1,3E+0</b>	7,9E-1	9,9E-1	<b>1,1E+0</b>	1,1E+0	<b>1,9E-1</b>	1,9E+0	1,8E+0
10 (M.)	<b>1,3E+1</b>	8,9E+1	<b>3,9E+1</b>	8,1E+1	8,0E+1	<b>1,1E+1</b>	8,9E+1	<b>6,1E+0</b>	8,0E+1	8,2E+1
(DP)	<b>2,3E+0</b>	3,8E-1	<b>1,9E+0</b>	1,9E+0	1,6E+0	<b>2,2E+0</b>	1,7E+0	<b>0,0E+0</b>	4,4E+0	2,7E+0
15 (M.)	<b>1,8E+1</b>	1,3E+2	<b>5,9E+1</b>	1,2E+2	1,2E+2	<b>1,7E+1</b>	1,3E+2	<b>9,0E+0</b>	1,2E+2	1,2E+2
(DP)	<b>2,1E+0</b>	9,6E-1	<b>1,9E+0</b>	2,8E+0	2,5E+0	<b>2,4E+0</b>	2,5E+0	<b>0,0E+0</b>	5,8E+0	2,8E+0
20 (M.)	<b>2,6E+1</b>	1,8E+2	<b>8,0E+1</b>	1,6E+2	1,6E+2	<b>2,4E+1</b>	1,7E+2	<b>1,2E+1</b>	1,6E+2	1,7E+2
(DP)	<b>3,7E+0</b>	9,7E-1	<b>3,1E+0</b>	3,6E+0	3,6E+0	<b>5,1E+0</b>	3,2E+0	<b>0,0E+0</b>	7,7E+0	4,5E+0
DTLZ7 #Obj.	Spacing					Hypervolume				
	RefPar	RefDec	Bpar	Bdec	AsBDec	RefPar	RefDec	Bpar	Bdec	AsBDec
2 (M.)	2,4E+0	<b>5,2E-2</b>	1,8E+0	2,5E-1	1,9E-1	<b>1,6E+1</b>	5,5E-1	<b>1,5E+1</b>	1,3E+0	2,8E+0
(DP)	6,0E-1	<b>2,4E-2</b>	4,1E-1	1,1E-1	1,1E-1	<b>5,0E-1</b>	4,6E-1	<b>1,4E+0</b>	4,7E-1	1,1E+0
3 (M.)	3,2E+0	<b>9,4E-2</b>	4,4E+0	<b>5,6E-1</b>	<b>1,8E-1</b>	<b>2,2E+1</b>	1,3E+0	<b>2,7E+1</b>	<b>2,8E+0</b>	<b>2,3E+0</b>
(DP)	4,8E-1	<b>4,7E-2</b>	7,8E-1	<b>2,0E-1</b>	<b>9,5E-2</b>	<b>3,6E-1</b>	5,9E-1	<b>2,1E-3</b>	<b>1,1E+0</b>	<b>1,2E+0</b>
5 (M.)	1,1E+0	<b>2,8E-1</b>	8,1E+0	1,2E+0	4,1E-1	<b>2,7E+1</b>	2,6E+0	<b>4,5E+1</b>	3,2E+0	1,9E+0
(DP)	2,4E-1	<b>1,7E-1</b>	1,2E+0	4,2E-1	2,1E-1	<b>4,9E+0</b>	1,1E+0	<b>8,7E-3</b>	2,0E+0	1,4E+0
10 (M.)	1,6E+0	<b>8,6E-1</b>	1,2E+1	<b>1,6E+0</b>	<b>5,7E-1</b>	5,2E+0	2,5E+0	<b>9,0E+1</b>	6,4E-1	2,0E-1
(DP)	3,2E-1	<b>1,8E-1</b>	1,5E+0	<b>8,3E-1</b>	<b>3,4E-1</b>	3,7E+0	1,1E+0	<b>0,0E+0</b>	6,4E-1	1,6E-1
15 (M.)	2,1E+0	1,5E+0	1,8E+1	2,0E+0	<b>6,7E-1</b>					
(DP)	3,7E-1	3,2E-1	1,2E+0	1,3E+0	<b>3,8E-1</b>					
20 (M.)	2,6E+0	2,0E+0	2,4E+1	2,0E+0	<b>8,9E-1</b>					
(DP)	5,1E-1	3,8E-1	2,0E+0	1,4E+0	<b>5,5E-1</b>					

cador apontou um bom desempenho para o algoritmo RefDec, isto é, uma distribuição equilibrada ao longo das fronteiras. Porém todos os outros indicadores apontam que o referido algoritmo não conseguiu convergir suas soluções em direção à fronteira de Pareto real. As fronteiras obtidas não aproximaram-se da fronteira real.

## Uma síntese empírica baseada na análise dos resultados

Ao avaliar os resultados obtidos, uma constatação contundente é: não existe uma única estratégia que apresente os melhores resultados para todas as classes de problemas.

Desta forma, podemos interpretar situações onde determinadas abordagens mostram-se mais adequadas em relação a outras. Tais situações são sumarizadas a seguir:

- algoritmos baseados em decomposição mostram-se mais aptos em “escapar” de ótimos locais. Até 3 objetivos, manter uma única população é uma boa estratégia. Porém, para mais objetivos, a execução distribuída apresenta desempenho superior;

- em relação à escalabilidade de um MOEA, para a maioria dos problemas, a execução distribuída de algoritmos baseados em decomposição apresenta melhores resultados;
- problemas que investigam a capacidade de um algoritmo gerar soluções bem distribuídas apontam que as estratégias distribuídas são mais adequadas. Entretanto, quando avalia-se especificamente a convergência das soluções, o uso de uma estratégia centralizada baseada em decomposição mostra-se uma opção mais pertinente;
- modificações em um mesmo problema base pode privilegiar estratégias distintas. Estratégias centralizadas apresentam desempenho superior na DTLZ3, enquanto que estratégias distribuídas são melhores na DTLZ5. Ambas utilizam a DTLZ2 como problema base;
- de maneira geral, os algoritmos baseados em decomposição apresentam desempenho competitivo para a maioria dos problemas propostos. Porém, as estratégias baseadas em relações de dominância entre as soluções são melhores em manter indivíduos em diferentes regiões de uma fronteira de Pareto real desconexa. O problema DTLZ7 ilustra esse tipo de problema e, especificamente para este cenário, partículas distribuídas em diferentes enxames mostra-se uma estratégia mais adequada.

## **CAPÍTULO 7**

### **CONCLUSÃO**

Neste trabalho, utilizamos múltiplos enxames para resolver problemas de otimização. Inicialmente conduzimos experimentos para validar nossa estratégia distribuída considerando a otimização de problemas com um objetivo. O impacto da topologia entre os enxames foi investigado, considerando que enxames independentes compartilharam suas experiências assincronamente. O número de enxames foi variado e diferentes topologias e estratégias de comunicação foram analisadas. Confrontamos os resultados gerados aos obtidos por um modelo de otimização centralizado. Os experimentos demonstram a influência exercida pela relação entre subpopulações independentes na otimização por enxame de partículas.

Inicialmente aplicada à resolução de problemas mono-objetivo, o algoritmo PSO é uma técnica utilizada para a resolução de problemas com mais de um objetivo. Isso se deve, entre outros fatores, à sua codificação relativamente simples, envolvendo parâmetros adaptáveis e por ser uma meta-heurística baseada em populações. Esta característica é particularmente relevante em problemas que possuem um conjunto de soluções e não apenas uma única solução possível.

Todavia, apesar da intensa pesquisa existente tendo a otimização por enxame de partículas como foco, várias lacunas demandam estudos aprofundados, principalmente envolvendo o uso de múltiplos enxames para a resolução de problemas com vários objetivos. Neste contexto, surgiram diversas questões associadas à execução cooperativa de algoritmos PSO aplicados a problemas com mais de um objetivo, sobretudo com vários objetivos.

Neste trabalho, foram propostas estratégias distribuídas baseadas no PSO com múltiplos e independentes enxames para resolver problemas com vários objetivos. Partículas foram divididas em subpopulações independentes e cada enxame manteve

um “*buffer*” que armazenou os melhores indivíduos de cada subpopulação. Métodos assíncronos de comunicação foram estabelecidos, baseados na melhoria da aptidão das partículas. Estes métodos permitiram o compartilhamento de indivíduos entre enxames para ajudar a explorar regiões promissoras do espaço de busca.

Outra proposta explorada no trabalho é o uso da técnica de decomposição para a resolução de problemas com vários objetivos, para a qual foram projetados dois modelos distribuídos - síncrono e assíncrono. Os problemas multiobjetivo foram decompostos em vários subproblemas mono-objetivos, de modo a permitir que valores de aptidão fossem obtidos em função de cada subproblema.

Finalmente, utilizamos testes de *benchmarks* tradicionais e confrontamos os resultados obtidos com a estratégia tradicional, na qual o algoritmo PSO é baseado em população centralizada. Foram considerados os métodos baseados na decomposição de MOPs e na relação de dominância entre soluções. A partir da observação dos resultados experimentais foi possível concluir que a execução distribuída de vários enxames tem um efeito positivo sobre a convergência e diversidade do processo de otimização para problemas com vários objetivos. A análise de diferentes problemas permitiu determinar qual estratégia de otimização baseada no algoritmo PSO é mais indicada tendo em vista as características específicas do problema a ser resolvido.

Trabalhos futuros incluem: o estudo de outras estruturas de comunicação para a execução paralela e distribuída. Tais estruturas podem abordar redes ponto-a-ponto, plataformas de computação paralela (incluindo múltiplos processadores/núcleos e GPU - *Graphics Processing Unit*), entre outras. Analisar o desempenho da otimização ao ajustar dinamicamente o tamanho da população de cada enxame. Variar dinamicamente o número de subpopulações do processo cooperativo. Implementar múltiplos enxames onde cada subpopulação utiliza diferentes valores associados às constantes empregadas na equação de atualização da velocidade da partícula. Outro estudo interessante está relacionado ao estabelecimento de enxames especializados em cobrir partes da fronteira de Pareto, para promover a convergência e explorar a fronteira em sua totalidade.

## BIBLIOGRAFIA

- [Al Moubayed et al., 2012] Al Moubayed, N., Petrovski, A., e McCall, J. (2012). D2MOPSO: multi-objective particle swarm optimizer based on decomposition and dominance. **EvoCOP'12. Proceedings of the 12th European conference on Evolutionary Computation in Combinatorial Optimization**. Malaga, Espanha, 2012. p. 75–86.
- [Angeline, 1998] Angeline, P. (1998). Using selection to improve particle swarm optimization. **Evolutionary Computation Proceedings, 1998. IEEE International Conference on IEEE World Congress on Computational Intelligence..** maio. 1998. p. 84 –89.
- [Baker et al., 1972] Baker, B., Shostak, R., Bumby, I. R. T., Hajnal, A., Milner, E. C., e Szemerédi, E. (1972). Gossips and telephones. **Discrete Mathematics**, v.2, p. 191–193, 1972.
- [Bentley e Wakefield, 1998] Bentley, P. J. e Wakefield, J. P. (1998). Finding acceptable solutions in the pareto-optimal range using multiobjective genetic algorithms. CHAUDHRY, P. K.; ROY, R.; PANT, R. K., editores, **Soft Computing in Engineering Design and Manufacturing**, p. 231–240. Heidelberg, Alemanha, 1998.
- [Biazzini e Montresor, 2010] Biazzini, M. e Montresor, A. (2010). Gossiping differential evolution: A decentralized heuristic for function optimization in p2p networks. **2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)**. Xangai, China, 2010. p. 468 –475.
- [Bona et al., 2008] Bona, L., Fonseca, K., Duarte, E., e de Mello, S. (2008). Hyperbone: A scalable overlay network based on a virtual hypercube. **8th IEEE International Symposium on Cluster Computing and the Grid, 2008. CCGRID '08**. Lyon, França, 2008. p. 58–64.

- [Branke et al., 2008] Branke, J., Deb, K., Miettinen, K., e Slowinski, R., editores (2008). **Multiobjective Optimization, Interactive and Evolutionary Approaches [outcome of Dagstuhl seminars]**., v. 5252 da Lecture Notes in Computer Science. Springer, 2008.
- [Branke e Mostaghim, 2006] Branke, J. e Mostaghim, S. (2006). About selecting the personal best in multi-objective particle swarm optimization. Editors RUNARSSON, T. P.; BEYER, H.-G.; BURKE, E. K.; GUERVÓS, J. J. M.; WHITLEY, L. D.; YAO, X. RUNARSSON, T. P.; BEYER, H.-G.; BURKE, E. K.; GUERVÓS, J. J. M.; WHITLEY, L. D.; YAO, X., editores, **Parallel Problem Solving from Nature – PPSN**. v. 4193 da Lecture Notes in Computer Science. Springer, 2006. p. 523–532.
- [Britto e Pozo, 2012] Britto, A. e Pozo, A. (2012). Using archiving methods to control convergence and diversity for many-objective problems in particle swarm optimization. **2012 IEEE Congress on Evolutionary Computation (CEC)**. Brisbane, Australia, 2012. p. 1–8.
- [Campos Jr. et al., 2011] Campos Jr., A. d., Pozo, A. T. R., e Duarte Jr., E. P. (2011). The impact of the topology on multiple swarms particle optimization using asynchronous communication. **WPABA'11: Proceedings of the Forth International Workshop on Parallel Architectures and Bioinspired Algorithms (WPABA 2011)**. Galveston, EUA, 2011. p. 11–20.
- [Castro et al., 2012] Castro, O., Britto, A., e Pozo, A. (2012). A comparison of methods for leader selection in many-objective problems. **2012 IEEE Congress on Evolutionary Computation (CEC)**. Brisbane, Australia, 2012. p. 1–8.
- [Chakraborty et al., 2010] Chakraborty, P., Das, S., Abraham, A., Snasel, V., e Roy, G. (2010). On convergence of multi-objective particle swarm optimizers. **2010 IEEE Congress on Evolutionary Computation (CEC)**. Barcelona, Espanha, 2010. p. 1–8.



- [Clerc, 1999] Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. **Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99.** v.3. Washington, EUA, 1999.
- [Coello, 1999] Coello, C. (1999). An updated survey of evolutionary multiobjective optimization techniques: state of the art and future trends. **Proceedings of the 1999 Congress on Evolutionary Computation, 1999..** v.1. Washington, EUA, 1999.
- [Coello e Cortés, 2005] Coello, C. A. C. e Cortés, N. C. (2005). Solving multiobjective optimization problems using an artificial immune system. **Genetic Programming and Evolvable Machines**, v.6, n. 2, p. 163–190, 2005.
- [Coello et al., 2007] Coello, C. C., Lamont, G., e van Veldhuizen, D. (2007). **Evolutionary Algorithms for Solving Multi-Objective Problems.** 2nd ed., Genetic and Evolutionary Computation. Londres, Inglaterra, 2007.
- [Coello Coello e Lechuga, 2002] Coello Coello, C. e Lechuga, M. (2002). Mopso: a proposal for multiple objective particle swarm optimization. **Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02.** v.2. Havaí, EUA, 2002. p. 1051 –1056.
- [Daneshyari e Yen, 2008] Daneshyari, W. e Yen, G. (2008). Cultural mopso: A cultural framework to adapt parameters of multiobjective particle swarm optimization. **Evolutionary Computation, 2008. CEC 2008. (IEEE Congress on IEEE World Congress on Computational Intelligence).** Hong Kong, China, 2008. p. 1325 –1332.
- [Das, 1999] Das, I. (1999). On characterizing the “knee” of the Pareto curve based on normal-boundary intersection. **Structural optimization**, v.18, n. 2-3, p. 107–115, 1999.
- [Das e Dennis, 1998] Das, I. e Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization

- problems. **SIAM J. on Optimization**, Philadelphia, EUA, v.8, n. 3, p. 631–657, março. 1998.
- [Dasgupta, 1997] Dasgupta, D. (1997). Artificial neural networks and artificial immune systems: similarities and differences. **Systems, Man, and Cybernetics, 1997. 1997 IEEE International Conference on Computational Cybernetics and Simulation**. v.1. Orlando, EUA, 1997. p. 873 –878.
- [de Britto et al., 2013] de Britto, A. B., Mostaghim, S., e Pozo, A. (2013). Iterated multi-swarm: a multi-swarm algorithm based on archiving methods. **Genetic and Evolutionary Computation Conference**. Amsterdã , Holanda, 2013. p. 583–590.
- [de Campos et al., 2013] de Campos, A., Pozo, A. T., e Duarte, E. P. (2013). Evaluation of asynchronous multi-swarm particle optimization on several topologies. **Concurrency and Computation: Practice and Experience**, 25, n. 8, p. 1057–1071, 2013.
- [de Carvalho e Pozo, 2010] de Carvalho, A. e Pozo, A. (2010). The control of dominance area in particle swarm optimization algorithms for many-objective problems. **2010 Eleventh Brazilian Symposium on Neural Networks (SBRN)**. São Bernardo do Campo, Brasil, 2010. p. 140–145.
- [de Carvalho e Pozo, 2011a] de Carvalho, A. B. e Pozo, A. (2011a). Measuring the convergence and diversity of CDAS multi-objective particle swarm optimization algorithms: A study of many-objective problems. **Neurocomputing**, v.75, p. 43–51, Amsterdã , Holanda, 2011.
- [de Carvalho e Pozo, 2011b] de Carvalho, A. B. e Pozo, A. (2011b). Using different many-objective techniques in particle swarm optimization for many objective problems: An empirical study. **International Journal of Computer Information System and Industrial Management Applications**, v.3, p. 96–197, 2011.
- [Deb, 2001] Deb, K. (2001). **Multi-objective optimization using evolutionary algorithms**. Chichester, New York: John Wiley & Sons, 2001.

- [Deb, 2011] Deb, K. (2011). Multi-objective optimization using evolutionary algorithms: An introduction. Relatório Técnico KanGAL Report No.2011003, Kanpur Genetic Algorithms Laboratory, Fevereiro. 2011.
- [Deb et al., 2002a] Deb, K., Pratap, A., Agarwal, S., e Meyarivan, T. (2002a). A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, 6, n. 2, p. 182–197, Havaí, EUA, 2002.
- [Deb et al., 2002b] Deb, K., Thiele, L., Laumanns, M., e Zitzler, E. (2002b). Scalable Multi-Objective Optimization Test Problems. **Congress on Evolutionary Computation (CEC 2002)**. IEEE Press, 2002. p. 825–830.
- [Demšar, 2006] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. **J. Mach. Learn. Res.**, 7, p. 1–30, dezembro. 2006.
- [Derrac et al., 2011] Derrac, J., García, S., Molina, D., e Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. **Swarm and Evolutionary Computation**, 1, n. 1, p. 3–18, 2011.
- [Dorigo et al., 1996] Dorigo, M., Maniezzo, V., e Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, 26, n. 1, p. 29 –41, fevereiro. 1996.
- [Doval e O'Mahony, 2003] Doval, D. e O'Mahony, D. (2003). Overlay networks: A scalable alternative for p2p. **Internet Computing, IEEE**, 7, n. 4, p. 79–82, 2003.
- [Durillo e Nebro, 2011] Durillo, J. J. e Nebro, A. J. (2011). jmetal: A java framework for multi-objective optimization. **Advances in Engineering Software**, 42, p. 760–771, 2011.
- [Eberhart e Shi, 2000] Eberhart, R. e Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. **Proceedings of the 2000 Congress on Evolutionary Computation, 2000**. v.1. 2000. p. 84 –88.

- [Mullender, 1993] (editor), S. M. (1993). **Distributed Systems**. 2nd ed. Addison-Wesley, 1993.
- [El-Abd e Kamel, 2008] El-Abd, M. e Kamel, M. S. (2008). A taxonomy of cooperative particle swarm optimizers. **International Journal of Computational Intelligence Research**, 4, n. 2, p. 137–144, 2008.
- [Eugster et al., 2004] Eugster, P., Guerraoui, R., Kermarrec, A.-M., e Massoulie, L. (2004). Epidemic information dissemination in distributed systems. **Computer**, 37, n. 5, p. 60 – 67, maio.. 2004.
- [Fdhila et al., 2011] Fdhila, R., Hamdani, T., e Alimi, A. (2011). Distributed mopso with a new population subdivision technique for the feature selection. **2011 5th International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII)**. Budapeste, Hungria, 2011. p. 81 –86.
- [Fonseca e Fleming, 1993] Fonseca, C. e Fleming, P. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. **Proceedings of the 5th International Conference on Genetic Algorithms**. Urbana-Champaign, EUA, 1993. p. 416–423.
- [García et al., 2009] García, S., Molina, D., Lozano, M., e Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. **Journal of Heuristics**, Hingham, MA, USA, 15, n. 6, p. 617–644, dezembro. 2009.
- [Guo et al., 2010] Guo, P., Wang, X., e Han, Y. (2010). The enhanced genetic algorithms for the optimization design. **2010 3rd International Conference on Biomedical Engineering and Informatics (BMEI)**. v.7, Yantai, China, 2010. p. 2990 –2994.
- [Hedetniemi et al., 1988] Hedetniemi, S. M., Hedetniemi, S. T., e Liestman, A. L. (1988). A survey of gossiping and broadcasting in communication networks. **Networks**, 18, n. 4, p. 319–349, 1988.

- [Hong-Xia et al., 2008] Hong-Xia, Y., Liu, X., e Pu-Kun, L. (2008). Intelligent algorithms for solving multiobjective optimization problems. **4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08**. Dalian, China, 2008. p. 1 –5.
- [Hu e Yen, 2013] Hu, W. e Yen, G. (2013). Density estimation for selecting leaders and mantaining archive in MOPSO. **2013 IEEE Congress on Evolutionary Computation (CEC)**. Cancún, México, 2013. p. 181–188.
- [Hu e Eberhart, 2002] Hu, X. e Eberhart, R. (2002). Multiobjective optimization using dynamic neighborhood particle swarm optimization. **2002. CEC '02. Proceedings of the 2002 Congress on Evolutionary Computation**. v.2, Havaí, EUA, 2002. p. 1677 –1681.
- [Ishibuchi et al., 2009] Ishibuchi, H., Sakane, Y., Tsukamoto, N., e Nojima, Y. (2009). Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. **IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009**. San Antonio, EUA, 2009. p. 1758 –1763.
- [Ishibuchi et al., 2008] Evolutionary many-objective optimization: A short review. **IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence)**. Hong Kong, China, 2008. p. 2419 –2426.
- [Jaimes e Coello, 2009] Jaimes, A. L. e Coello, C. A. C. (2009). Some techniques to deal with many-objective problems. **Genetic and Evolutionary Computation Conference -GECCO**. Montreal, Canadá, 2009. p. 2693–2696.
- [Ke et al., 2013] Ke, L., Zhang, Q., e Battiti, R. (2013). MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony, 2013.
- [Kennedy, 1999] Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. **Evolutionary Computation**,

**1999. CEC 99. Proceedings of the 1999 Congress on.** v.3, Washington, EUA, 1999.

[Kennedy e Eberhart, 1995] Kennedy, J. e Eberhart, R. (1995). Particle swarm optimization. **IEEE International Conference on Neural Networks, 1995.** v.4, Perth, Austrália, 1995. p. 1942 –1948 vol.4.

[Kohonen, 1995] Kohonen, T. (1995). **Self-Organizing Maps.** Berlin; New York: Springer, 1995.

[Kshemkalyani e Singhal, 2011] Kshemkalyani, A. D. e Singhal, M. (2011). **Distributed Computing: Principles, Algorithms, and Systems.** Cambridge, UK: Cambridge University Press, 2011.

[Laumanns et al., 2002] Laumanns, M., Thiele, L., Deb, K., e Zitzler, E. (2002). Combining convergence and diversity in evolutionary multi-objective optimization. **Evolutionary Computation**, v.10, 2002.

[Lee e Kim, 2011] Lee, K.-B. e Kim, J.-H. (2011). Multi-objective particle swarm optimization with preference-based sorting. **2011 IEEE Congress on Evolutionary Computation (CEC).** Nova Orleans, EUA, 2011. p. 2506 –2513.

[Li e Xiao, 2008] Li, J. e Xiao, X. (2008). Multi-swarm and multi-best particle swarm optimization algorithm. **7th World Congress on Intelligent Control and Automation, 2008. WCICA 2008.** Chongqing, China, 2008. p. 6281 –6286.

[Li, 2004] Li, X. (2004). Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function. Em Deb, K., Poli, R., Banzhaf, W., Beyer, H.-G., Burke, E. K., Darwen, P. J., Dasgupta, D., Floreano, D., Foster, J. A., Harman, M., Holland, O., Lanzi, P. L., Spector, L., Tettamanzi, A., Thierens, D., e Tyrrell, A. M., editores, *GECCO (1)*, volume 3102 de *Lecture Notes in Computer Science*, pgs. 117–128. Springer.

- [Li e Yao, 2009] Li, X. e Yao, X. (2009). Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms. **IEEE Congress on Evolutionary Computation, 2009. CEC '09**. Trondheim, Noruega, 2009. p. 1546 –1553.
- [Li e Yao, 2011] Li, X. e Yao, X. (2011). Cooperatively coevolving particle swarms for large scale optimization. **IEEE Transactions on Evolutionary Computation**, p. 1, 2011.
- [Li et al., 2010] Li, Y., Liang, J., e Hu, J. (2010). A multi-swarm cooperative hybrid particle swarm optimizer. **2010 Sixth International Conference on Natural Computation (ICNC)**. v.5. Yantai, China, 2010. p. 2535 –2539.
- [Liang e Suganthan, 2005] Liang, J. e Suganthan, P. (2005). Dynamic multi-swarm particle swarm optimizer. **Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE**. Pasadena, EUA, 2005. p. 124 – 129.
- [Liang et al., 2007] Liang, O., Sekercioglu, Y., e Mani, N. (2007). A low-cost flooding algorithm for wireless sensor networks. **Wireless Communications and Networking Conference. WCNC 2007**. Hong Kong, China, 2007. p. 3495 –3500.
- [Liu et al., 2008] Liu, X., Fang, Z., Si, H., Yi, T., e Chen, K. (2008). Research of direction-based search algorithm in p2p network. **4th International Conference on Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08**. Dalian, China, 2008. p. 1–4.
- [López-Ibáñez et al., 2011] López-Ibáñez, M., Knowles, J., e Laumanns, M. (2011). On sequential online archiving of objective vectors. EMO'11. **Proceedings of the 6th international conference on Evolutionary multi-criterion optimization**. Ouro Preto, Brasil, 2011. p. 46–60.
- [Luke, 2009] Luke, S. (2009). **Essentials of Metaheuristics**. Lulu, 2009. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>.

- [LÃ\_vbjerg et al., 2001] LÃ\_vbjerg, M., Rasmussen, T. K., e Krink, T. (2001). Hybrid particle swarm optimiser with breeding and subpopulations. **Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)**. São Francisco, EUA, 2001. p. 469–476.
- [MacDougall, 1987] MacDougall, M. H. (1987). **Simulating computer systems: techniques and tools**. Cambridge, MA, USA: MIT Press, 1987.
- [Messac et al., 2003] Messac, A., Ismail-Yahaya, A., e Mattson, C. A. (2003). The normalized normal constraint method for generating the Pareto frontier. **Structural and Multidisciplinary Optimization**, 25, n. 2, p. 86–98, julho. 2003.
- [Miettinen, 1999] Miettinen, K. (1999). **Nonlinear multiobjective optimization**. Boston, USA: Kluwer, 1999.
- [Mitchell, 1998] Mitchell, M. (1998). **An Introduction to Genetic Algorithms**. Cambridge, MA, USA: MIT Press, 1998.
- [Montes de Oca et al., 2009] Montes de Oca, M., Stutzle, T., Birattari, M., e Dorigo, M. (2009). Frankenstein’s pso: A composite particle swarm optimization algorithm. **IEEE Transactions on Evolutionary Computation**, v.13, n. 5, p. 1120 –1132, outubro. 2009.
- [Mostaghim e Teich, 2003] Mostaghim, S. e Teich, J. (2003). Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). **Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS '03**. Indianapolis, EUA, 2003. p. 26 – 33.
- [Moubayed et al., 2010] Moubayed, N. A., Petrovski, A., e McCall, J. A. W. (2010). A novel smart multi-objective particle swarm optimisation using decomposition. **Parallel Problem Solving from Nature - PPSN**. Cracóvia, Polônia, 2010. p. 1–10.
- [Nebro et al., 2009] Nebro, A. J., Durillo, J. J., García-Nieto, J., Coello Coello, C. A., Luna, F., e Alba, E. (2009). SMPSO: A New PSO-based Metaheuristic for Multi-



- objective Optimization. **2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009)**. Nova Iorque, EUA, 2009. p. 66–73.
- [Obayashi e Sasaki, 2003] Obayashi, S. e Sasaki, D. (2003). Visualization and data mining of pareto solutions using self-organizing map. EMO'03. **Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization**. Faro, Portugal, 2003. p. 796–809.
- [Padhye, 2009] Padhye, N. (2009). Empirical comparison of mopso methods - guide selection and diversity preservation -. **IEEE Congress on Evolutionary Computation, 2009. CEC '09**. Trondheim, Noruega, 2009. p. 2516 –2523.
- [Padhye et al., 2009] Padhye, N., Branke, J., e Mostaghim, S. (2009). Comparison of archiving methods in multi-objective particle swarm optimization (mopso): empirical study. **Genetic and Evolutionary Computation Conference - GECCO**. Montreal, Canadá, 2009. p. 1755–1756.
- [Parra et al., 2009] Parra, F., Galan, S. G., Yuste, A. J., Prado, R. P., e Muñoz, J. E. (2009). A method to minimize distributed pso algorithm execution time in grid computer environment. IWINAC '09. **Proceedings of the 3rd International Work-Conference on The Interplay Between Natural and Artificial Computation: Part II: Bioinspired Applications in Artificial and Natural Computation**. Santiago de Compostela, Espanha, 2009. p. 478–487.
- [Parsopoulos e Vrahatis, 2002] Parsopoulos, K. e Vrahatis, M. (2002). Recent approaches to global optimization problems through particle swarm optimization. **Natural Computing**, v.1, n. 2-3, p. 235–306, 2002.
- [Parsopoulos, 2009] Parsopoulos, K. E. (2009). Cooperative micro-particle swarm optimization. GEC '09. **Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation**. Montreal, Canadá, 2009. p. 467–474.
- [Peng e Zhang, 2008] Peng, W. e Zhang, Q. (2008). A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization pro-

- blems. **IEEE International Conference on Granular Computing, 2008. GrC 2008.** Hangzhou, China, 2008. p. 534–537.
- [Potter e Jong, 1994] Potter, M. A. e Jong, K. A. D. (1994). A cooperative coevolutionary approach to function optimization. PPSN III. **Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature.** Jerusalem, Israel, 1994. p. 249–257.
- [Pryke et al., 2006] Pryke, A., Mostaghim, S., e Nazemi, A. (2006). Heatmap visualization of population based multi objective algorithms. Editors OBAYASHI, S.; DEB, K.; POLONI, C.; HIROYASU, T.; MURATA, T. OBAYASHI, S.; DEB, K.; POLONI, C.; HIROYASU, T.; MURATA, T., editores, **EMO**. v.4403 da Lecture Notes in Computer Science. Springer, 2006. p. 361–375.
- [R Development Core Team, 2011] R Development Core Team (2011). **R: A Language and Environment for Statistical Computing.** R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
- [Rakitianskaia e Engelbrecht, 2008] Rakitianskaia, A. e Engelbrecht, A. (2008). Cooperative charged particle swarm optimiser. **IEEE Congress on Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence).** Hong Kong, China, 2008. p. 933 –939.
- [Raquel e Naval, 2005] Raquel, C. R. e Naval, Jr., P. C. (2005). An effective use of crowding distance in multiobjective particle swarm optimization. GECCO '05. **Proceedings of the 2005 conference on Genetic and evolutionary computation.** Washington, EUA, 2005. p. 257–264.
- [Reyes-sierra e Coello, 2006] Reyes-sierra, M. e Coello, C. A. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. **International Journal of Computational Intelligence Research**, 2, n. 3, p. 287–308, 2006.

- [Saad e Schultz, 1988] Saad, Y. e Schultz, M. (1988). Topological properties of hypercubes. **IEEE Transactions on Computers**, 37, n. 7, p. 867–872, 1988.
- [Sandhu, 1997] Sandhu, S. (1997). **Development of a standard framework for manufacturing simulators**. Tese de doutorado. Coventry, Inglaterra, 1997.
- [Sato et al., 2007] Sato, H., Aguirre, H. E., e Tanaka, K. (2007). Controlling dominance area of solutions and its impact on the performance of MOEAs. EMO'07. **Proceedings of the 4th international conference on Evolutionary multi-criterion optimization**. Matsushima, Japão, 2007. p. 5–20.
- [Schott, 1995] Schott, J. R. (1995). **Fault Tolerant Design Using Single and Multi-criteria Genetic Algorithm Optimization**. Tese de Doutorado, 1995.
- [Schütze et al., 2012] Schütze, O., Esquivel, X., Lara, A., e Coello, C. A. C. (2012). Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. **IEEE Transaction on Evolutionary Computation**, v.16, n. 4, p. 504–522, 2012.
- [Schutze et al., 2011] Schutze, O., Lara, A., e Coello Coello, C. A. (2011). On the influence of the number of objectives on the hardness of a multiobjective optimization problem. **IEEE Transactions on Evolutionary Computation**, v.15, n. 4, p. 444–455, 2011.
- [Shi e Eberhart, 1998] Shi, Y. e Eberhart, R. (1998). A modified particle swarm optimizer. **The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence**. Anchorage, EUA, 1998. p. 69 –73.
- [Sierra e Coello, 2005] Sierra, M. R. e Coello, C. A. C. (2005). Improving pso-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance. **In EMO2005, pages 505–519. LNCS 3410**. Guanajuato, Mexico, 2005. p. 505–519.

- [Sinha et al., 2012] Sinha, A., Pandey, A., e Deb, K. (2012). Solving high objective problems in fixed interactions with the decision maker. **2012 IEEE Congress on Evolutionary Computation (CEC)**. Brisbane, Austrália, 2012. p. 1–8.
- [Srinivas e Deb, 1994] Srinivas, N. e Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary Computing**, Cambridge, MA, USA, 2, p. 221–248, setembro. 1994.
- [Suganthan et al., 2005] Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y., Auger, A., e Tiwari, S. (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. **Natural Computing**, p. 1–50, 2005.
- [Tang et al., 2007] Tang, K., Yao, X., Suganthan, P. N., MacNish, C., Chen, Y. P., Chen, C. M., e Yang, Z. (2007). Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Relatório técnico, Nature Inspired Computation and Applications Laboratory, USTC, China, 2007.
- [Teodorovic e Dell'orco, 2005] Teodorovic, D. e Dell'orco, M. (2005). Bee colony optimization - a cooperative learning approach to complex transportation problems. **Proceedings of the 10th EWGT Meeting and 16th Mini-EURO Conference**. Poznan, Polônia, 2005.
- [Thiele et al., 2009] Thiele, L., Miettinen, K., Korhonen, P. J., e Molina, J. (2009). A preference-based evolutionary algorithm for multi-objective optimization. **Evolutionary Computing**, Cambridge, MA, USA, 17, n. 3, p. 411–436, setembro. 2009.
- [Tomassini e Vanneschi, 2009] Tomassini, M. e Vanneschi, L. (2009). Introduction: special issue on parallel and distributed evolutionary algorithms, part I. **Genetic Programming and Evolvable Machines**, v.10, n. 4, p. 339–341, 2009.
- [Van Renesse et al., 1998] Van Renesse, R., Minsky, Y., e Hayden, M. (1998). A gossip-style failure detection service. Relatório técnico, Ithaca, NY, USA, 1998.

- [Van Veldhuizen e Lamont, 2002] Van Veldhuizen, D. A. e Lamont, G. B. (2002). On measuring multiobjective evolutionary algorithm performance. v.1. 2002. p. 204–211.
- [Vanneschi et al., 2010] Vanneschi, L., Codecasa, D., e Mauri, G. (2010). An empirical comparison of parallel and distributed particle swarm optimization methods. **Proceedings of the 12th annual conference on Genetic and evolutionary computation - GECCO**. Nova Iorque, EUA, 2010. p. 15–22.
- [Venske et al., 2012] Venske, S., Gonçalves, R., e Delgado, M. (2012). ADEMO/D: Adaptive differential evolution for multiobjective problems. **2012 Brazilian Symposium on Neural Networks (SBRN)**. Curitiba, Brasil, 2012. p. 226–231.
- [Wagner et al., 2007] Wagner, T., Beume, N., e Naujoks, B. (2007). Pareto-, aggregation-, and indicator-based methods in many-objective optimization. Em Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., e Murata, T., editores, *Evolutionary Multi-Criterion Optimization*, volume 4403 de *Lecture Notes in Computer Science*, p. 742–756. Matsushima, Japão, 2007.
- [Walker et al., 2013] Walker, D., Everson, R., e Fieldsend, J. (2013). Visualizing mutually nondominating solution sets in many-objective optimization. **IEEE Transactions on Evolutionary Computation**, v.17, n. 2, p. 165–184, 2013.
- [Wang et al., 2013] Wang, R., Zhang, T., e Guo, B. (2013). An enhanced MOEA/D using uniform directions and a pre-organization procedure. **2013 IEEE Congress on Evolutionary Computation (CEC)**. Cancún, México, 2013. p. 2390–2397.
- [Xiangwei e Hong, 2011] Xiangwei, Z. e Hong, L. (2011). A cooperative dual-swarm pso for dynamic optimization problems. **2011 Seventh International Conference on Natural Computation (ICNC)**. v.2. Havaí, EUA, 2011. p. 1131 –1135.
- [Yadong et al., 2010] Yadong, G., Jiye, H., Zhiliang, D., Shengnian, W., e Shaojun, H. (2010). Improved flooding-based resource discovery. maio. 2010. p. 1 –4.

- [Yan et al., 2007] Yan, J., Li, C., Wang, Z., Deng, L., e Sun, D. (2007). Diversity metrics in multi-objective optimization: Review and perspective. **IEEE International Conference on Integration Technology, 2007. ICIT '07**. Charlotte, EUA, 2007. p. 553–557.
- [Yang e Ding, 2007] Yang, Q. e Ding, S. (2007). Novel algorithm to calculate hypervolume indicator of Pareto approximation set. HUANG, D.-S.; HEUTTE, L.; LOOG, M., editores, **Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques**, v.2, Communications in Computer and Information Science, p. 235–244. Springer Berlin Heidelberg, 2007.
- [Zapotecas Martínez e Coello Coello, 2011] Zapotecas Martínez, S. e Coello Coello, C. A. (2011). A multi-objective particle swarm optimizer based on decomposition. GECCO '11. **Proceedings of the 13th annual conference on Genetic and evolutionary computation**. Dublin, Irlanda, 2011. p. 69–76.
- [Zhang, 2011] Zhang, H. (2011). An analysis of multiple particle swarm optimizers with inertia weight with diversive curiosity. **2011 IEEE Congress on Evolutionary Computation (CEC)**. Nova Orleans, EUA, 2011. p. 1754–1761.
- [Zhang e Li, 2007] Zhang, Q. e Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on Evolutionary Computation**, v.11, n. 6, p. 712–731, 2007.
- [Zheng et al., 2007] Zheng, X., Chen, K., Lu, D., e Liu, H. (2007). A proposal for a cooperative coevolutionary pso. **First IEEE International Symposium on Information Technologies and Applications in Education, 2007. ISITAE '07**. Kunming, China, 2007. p. 324–329.
- [Zhongkai et al., 2010] Zhongkai, L., Zhencai, Z., e Huiqin, Z. (2010). DSMOPSO: A distance sorting based multiobjective particle swarm optimization algorithm. **2010**

**Sixth International Conference on Natural Computation (ICNC).** v.5, Yantai, China, 2010. p. 2749 –2753.

[Zitzler, 1999] Zitzler, E. (1999). **Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.** Tese de Doutorado, ETH Zurich, Switzerland, 1999.

[Zitzler et al., 2000] Zitzler, E., Deb, K., e Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. **Evolutionary Computation**, v.8, p. 173–195, 2000.

[Zitzler et al., 2003] Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., e da Fonseca, V. (2003). Performance assessment of multiobjective optimizers: an analysis and review. **IEEE Transactions on Evolutionary Computation**, v.7, n. 2, p. 117 – 132, abril. 2003.

## APÊNDICE A

### TESTES ESTATÍSTICOS

Neste Apêndice, são apresentados os testes estatísticos relativos aos resultados obtidos. Tais resultados são referentes aos algoritmos avaliados na Seção 6.4 do Capítulo 6.

No decorrer deste Apêndice, os algoritmos abordados são identificados da seguinte forma:

- RefPar: algoritmo sequencial de referência, baseado na noção de dominância entre soluções;
- RefDec: algoritmo sequencial de referência, baseado na decomposição do MOP em subproblemas mono-objetivo;
- BPar: algoritmo distribuído, baseado na noção de dominância entre soluções utilizando a topologia *broadcast* entre os enxames;
- BDec: versão síncrona do algoritmo distribuído, baseado na decomposição do MOP em subproblemas mono-objetivo, utilizando a topologia *broadcast* entre os enxames;
- AsBDec: versão assíncrona do algoritmo distribuído, baseado na decomposição do MOP em subproblemas mono-objetivo, utilizando a topologia *broadcast* entre os enxames.

Em relação às múltiplas comparações dos testes estatísticos, as identificações 1, 2, 3, 4 e 5 são referentes aos algoritmos RefPar, RefDec, BPar, BDec e AsBDec, respectivamente. Os testes estatísticos apresentados são correspondentes às saídas geradas pela ferramenta *R Project for Statistical Computing* [R Development Core Team, 2011].

#### A.1 Múltiplas comparações

##### DTLZ1 - 2 Objetivos

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	71	34.379	TRUE
1-3	76	34.379	TRUE
1-4	26	34.379	FALSE
1-5	77	34.379	TRUE
2-3	5	34.379	FALSE
2-4	45	34.379	TRUE
2-5	6	34.379	FALSE
3-4	50	34.379	TRUE
3-5	1	34.379	FALSE
4-5	51	34.379	TRUE

IGDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	57	34.379	TRUE
1-3	78	34.379	TRUE
1-4	5	34.379	FALSE
1-5	30	34.379	FALSE
2-3	21	34.379	FALSE
2-4	62	34.379	TRUE
2-5	27	34.379	FALSE
3-4	83	34.379	TRUE
3-5	48	34.379	TRUE
4-5	35	34.379	TRUE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
--	---------	--------------	------------



1-2	79	34.379	TRUE
1-3	76	34.379	TRUE
1-4	30	34.379	FALSE
1-5	85	34.379	TRUE
2-3	3	34.379	FALSE
2-4	49	34.379	TRUE
2-5	6	34.379	FALSE
3-4	46	34.379	TRUE
3-5	9	34.379	FALSE
4-5	55	34.379	TRUE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	10.5	34.379	FALSE
1-3	24.5	34.379	FALSE
1-4	68.5	34.379	TRUE
1-5	49.0	34.379	TRUE
2-3	14.0	34.379	FALSE
2-4	79.0	34.379	TRUE
2-5	59.5	34.379	TRUE
3-4	93.0	34.379	TRUE
3-5	73.5	34.379	TRUE
4-5	19.5	34.379	FALSE

**DTLZ1 - 3 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	80	34.379	TRUE
1-3	21	34.379	FALSE
1-4	71	34.379	TRUE
1-5	58	34.379	TRUE
2-3	59	34.379	TRUE
2-4	9	34.379	FALSE
2-5	22	34.379	FALSE
3-4	50	34.379	TRUE
3-5	37	34.379	TRUE
4-5	13	34.379	FALSE

IGDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	61.5	34.379	TRUE
1-3	7.5	34.379	FALSE
1-4	76.5	34.379	TRUE
1-5	32.0	34.379	FALSE
2-3	54.0	34.379	TRUE
2-4	15.0	34.379	FALSE
2-5	29.5	34.379	FALSE
3-4	69.0	34.379	TRUE
3-5	24.5	34.379	FALSE
4-5	44.5	34.379	TRUE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	91.5	34.379	TRUE
1-3	18.0	34.379	FALSE
1-4	70.5	34.379	TRUE
1-5	67.5	34.379	TRUE
2-3	73.5	34.379	TRUE
2-4	21.0	34.379	FALSE
2-5	24.0	34.379	FALSE
3-4	52.5	34.379	TRUE
3-5	49.5	34.379	TRUE
4-5	3.0	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	12.5	34.379	FALSE
1-3	15.5	34.379	FALSE
1-4	64.5	34.379	TRUE
1-5	87.5	34.379	TRUE
2-3	28.0	34.379	FALSE
2-4	77.0	34.379	TRUE
2-5	100.0	34.379	TRUE
3-4	49.0	34.379	TRUE
3-5	72.0	34.379	TRUE
4-5	23.0	34.379	FALSE

**DTLZ1 - 5 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	47	34.379	TRUE
1-3	17	34.379	FALSE
1-4	65	34.379	TRUE
1-5	80	34.379	TRUE
2-3	64	34.379	TRUE
2-4	18	34.379	FALSE
2-5	33	34.379	FALSE
3-4	82	34.379	TRUE
3-5	97	34.379	TRUE
4-5	15	34.379	FALSE

IGDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	44.5	34.379	TRUE
1-3	43.0	34.379	TRUE
1-4	40.5	34.379	TRUE
1-5	14.5	34.379	FALSE
2-3	1.5	34.379	FALSE
2-4	85.0	34.379	TRUE
2-5	59.0	34.379	TRUE
3-4	83.5	34.379	TRUE
3-5	57.5	34.379	TRUE
4-5	26.0	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	103	34.379	TRUE
1-3	2	34.379	FALSE
1-4	53	34.379	TRUE
1-5	66	34.379	TRUE
2-3	105	34.379	TRUE
2-4	50	34.379	TRUE
2-5	37	34.379	TRUE
3-4	55	34.379	TRUE
3-5	68	34.379	TRUE
4-5	13	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	113.5	34.379	TRUE
1-3	44.5	34.379	TRUE
1-4	61.0	34.379	TRUE
1-5	81.0	34.379	TRUE
2-3	69.0	34.379	TRUE
2-4	52.5	34.379	TRUE
2-5	32.5	34.379	FALSE

3-4	16.5	34.379	FALSE
3-5	36.5	34.379	TRUE
4-5	20.0	34.379	FALSE

**DTLZ1 - 10 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	51	34.379	TRUE
1-3	5	34.379	FALSE
1-4	80	34.379	TRUE
1-5	79	34.379	TRUE
2-3	56	34.379	TRUE
2-4	29	34.379	FALSE
2-5	28	34.379	FALSE
3-4	85	34.379	TRUE
3-5	84	34.379	TRUE
4-5	1	34.379	FALSE

IGDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	41	34.379	TRUE
1-3	56	34.379	TRUE
1-4	33	34.379	FALSE
1-5	19	34.379	FALSE
2-3	15	34.379	FALSE
2-4	74	34.379	TRUE
2-5	60	34.379	TRUE
3-4	89	34.379	TRUE
3-5	75	34.379	TRUE
4-5	14	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	106	34.379	TRUE
1-3	2	34.379	FALSE
1-4	58	34.379	TRUE
1-5	64	34.379	TRUE
2-3	104	34.379	TRUE
2-4	48	34.379	TRUE
2-5	42	34.379	TRUE
3-4	56	34.379	TRUE
3-5	62	34.379	TRUE
4-5	6	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	111	34.379	TRUE
1-3	66	34.379	TRUE
1-4	54	34.379	TRUE
1-5	64	34.379	TRUE
2-3	45	34.379	TRUE
2-4	57	34.379	TRUE
2-5	47	34.379	TRUE
3-4	12	34.379	FALSE
3-5	2	34.379	FALSE
4-5	10	34.379	FALSE

**DTLZ1 - 15 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	57	34.379	TRUE

1-3	9	34.379	FALSE
1-4	84	34.379	TRUE
1-5	90	34.379	TRUE
2-3	48	34.379	TRUE
2-4	27	34.379	FALSE
2-5	33	34.379	FALSE
3-4	75	34.379	TRUE
3-5	81	34.379	TRUE
4-5	6	34.379	FALSE

IGDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	31	34.379	FALSE
1-3	43	34.379	TRUE
1-4	50	34.379	TRUE
1-5	24	34.379	FALSE
2-3	12	34.379	FALSE
2-4	81	34.379	TRUE
2-5	55	34.379	TRUE
3-4	93	34.379	TRUE
3-5	67	34.379	TRUE
4-5	26	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	104.5	34.379	TRUE
1-3	5.0	34.379	FALSE
1-4	58.5	34.379	TRUE
1-5	69.5	34.379	TRUE
2-3	99.5	34.379	TRUE
2-4	46.0	34.379	TRUE
2-5	35.0	34.379	TRUE
3-4	53.5	34.379	TRUE
3-5	64.5	34.379	TRUE
4-5	11.0	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	110	34.379	TRUE
1-3	65	34.379	TRUE
1-4	30	34.379	FALSE
1-5	65	34.379	TRUE
2-3	45	34.379	TRUE
2-4	80	34.379	TRUE
2-5	45	34.379	TRUE
3-4	35	34.379	TRUE
3-5	0	34.379	FALSE
4-5	35	34.379	TRUE

**DTLZ1 - 20 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	62	34.379	TRUE
1-3	13	34.379	FALSE
1-4	92	34.379	TRUE
1-5	83	34.379	TRUE
2-3	49	34.379	TRUE
2-4	30	34.379	FALSE
2-5	21	34.379	FALSE
3-4	79	34.379	TRUE
3-5	70	34.379	TRUE
4-5	9	34.379	FALSE

IGDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	22	34.379	FALSE
1-3	32	34.379	FALSE
1-4	53	34.379	TRUE
1-5	31	34.379	FALSE
2-3	10	34.379	FALSE
2-4	75	34.379	TRUE
2-5	53	34.379	TRUE
3-4	85	34.379	TRUE
3-5	63	34.379	TRUE
4-5	22	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	108	34.379	TRUE
1-3	6	34.379	FALSE
1-4	65	34.379	TRUE
1-5	61	34.379	TRUE
2-3	102	34.379	TRUE
2-4	43	34.379	TRUE
2-5	47	34.379	TRUE
3-4	59	34.379	TRUE
3-5	55	34.379	TRUE
4-5	4	34.379	FALSE

**DTLZ2 - 2 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	32	34.379	FALSE
1-3	64	34.379	TRUE
1-4	114	34.379	TRUE
1-5	90	34.379	TRUE
2-3	32	34.379	FALSE
2-4	82	34.379	TRUE
2-5	58	34.379	TRUE
3-4	50	34.379	TRUE
3-5	26	34.379	FALSE
4-5	24	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	65	34.379	TRUE
1-3	2	34.379	FALSE
1-4	102	34.379	TRUE
1-5	41	34.379	TRUE
2-3	63	34.379	TRUE
2-4	37	34.379	TRUE
2-5	24	34.379	FALSE
3-4	100	34.379	TRUE
3-5	39	34.379	TRUE
4-5	61	34.379	TRUE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	82	34.379	TRUE
1-3	120	34.379	TRUE
1-4	68	34.379	TRUE
1-5	30	34.379	FALSE
2-3	38	34.379	TRUE
2-4	14	34.379	FALSE

2-5	52	34.379	TRUE
3-4	52	34.379	TRUE
3-5	90	34.379	TRUE
4-5	38	34.379	TRUE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	57	34.379	TRUE
1-3	32	34.379	FALSE
1-4	82	34.379	TRUE
1-5	33	34.379	FALSE
2-3	89	34.379	TRUE
2-4	25	34.379	FALSE
2-5	24	34.379	FALSE
3-4	114	34.379	TRUE
3-5	65	34.379	TRUE
4-5	49	34.379	TRUE

**DTLZ2 - 3 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	40	34.379	TRUE
1-3	55	34.379	TRUE
1-4	106	34.379	TRUE
1-5	84	34.379	TRUE
2-3	15	34.379	FALSE
2-4	66	34.379	TRUE
2-5	44	34.379	TRUE
3-4	51	34.379	TRUE
3-5	29	34.379	FALSE
4-5	22	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	20	34.379	FALSE
1-3	19	34.379	FALSE
1-4	103	34.379	TRUE
1-5	73	34.379	TRUE
2-3	1	34.379	FALSE
2-4	83	34.379	TRUE
2-5	53	34.379	TRUE
3-4	84	34.379	TRUE
3-5	54	34.379	TRUE
4-5	30	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	97	34.379	TRUE
1-3	109	34.379	TRUE
1-4	56	34.379	TRUE
1-5	28	34.379	FALSE
2-3	12	34.379	FALSE
2-4	41	34.379	TRUE
2-5	69	34.379	TRUE
3-4	53	34.379	TRUE
3-5	81	34.379	TRUE
4-5	28	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	18	34.379	FALSE
1-3	67	34.379	TRUE

1-4	51	34.379	TRUE
1-5	1	34.379	FALSE
2-3	49	34.379	TRUE
2-4	69	34.379	TRUE
2-5	17	34.379	FALSE
3-4	118	34.379	TRUE
3-5	66	34.379	TRUE
4-5	52	34.379	TRUE

**DTLZ2 - 5 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	53	34.379	TRUE
1-3	39	34.379	TRUE
1-4	105	34.379	TRUE
1-5	103	34.379	TRUE
2-3	14	34.379	FALSE
2-4	52	34.379	TRUE
2-5	50	34.379	TRUE
3-4	66	34.379	TRUE
3-5	64	34.379	TRUE
4-5	2	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	47	34.379	TRUE
1-3	43	34.379	TRUE
1-4	114	34.379	TRUE
1-5	91	34.379	TRUE
2-3	4	34.379	FALSE
2-4	67	34.379	TRUE
2-5	44	34.379	TRUE
3-4	71	34.379	TRUE
3-5	48	34.379	TRUE
4-5	23	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	37	34.379	TRUE
1-3	61	34.379	TRUE
1-4	102	34.379	TRUE
1-5	100	34.379	TRUE
2-3	24	34.379	FALSE
2-4	65	34.379	TRUE
2-5	63	34.379	TRUE
3-4	41	34.379	TRUE
3-5	39	34.379	TRUE
4-5	2	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	30	34.379	FALSE
1-3	84	34.379	TRUE
1-4	21	34.379	FALSE
1-5	27	34.379	FALSE
2-3	54	34.379	TRUE
2-4	51	34.379	TRUE
2-5	3	34.379	FALSE
3-4	105	34.379	TRUE
3-5	57	34.379	TRUE
4-5	48	34.379	TRUE

**DTLZ2 - 10 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	49.0	34.379	TRUE
1-3	41.0	34.379	TRUE
1-4	102.5	34.379	TRUE
1-5	107.5	34.379	TRUE
2-3	8.0	34.379	FALSE
2-4	53.5	34.379	TRUE
2-5	58.5	34.379	TRUE
3-4	61.5	34.379	TRUE
3-5	66.5	34.379	TRUE
4-5	5.0	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	52	34.379	TRUE
1-3	45	34.379	TRUE
1-4	111	34.379	TRUE
1-5	87	34.379	TRUE
2-3	7	34.379	FALSE
2-4	59	34.379	TRUE
2-5	35	34.379	TRUE
3-4	66	34.379	TRUE
3-5	42	34.379	TRUE
4-5	24	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	109	34.379	TRUE
1-3	66	34.379	TRUE
1-4	64	34.379	TRUE
1-5	61	34.379	TRUE
2-3	43	34.379	TRUE
2-4	45	34.379	TRUE
2-5	48	34.379	TRUE
3-4	2	34.379	FALSE
3-5	5	34.379	FALSE
4-5	3	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	29	34.379	FALSE
1-3	72	34.379	TRUE
1-4	36	34.379	TRUE
1-5	0	34.379	FALSE
2-3	43	34.379	TRUE
2-4	65	34.379	TRUE
2-5	29	34.379	FALSE
3-4	108	34.379	TRUE
3-5	72	34.379	TRUE
4-5	36	34.379	TRUE

**DTLZ2 - 15 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	42	34.379	TRUE
1-3	48	34.379	TRUE
1-4	104	34.379	TRUE
1-5	106	34.379	TRUE
2-3	6	34.379	FALSE
2-4	62	34.379	TRUE
2-5	64	34.379	TRUE

3-4	56	34.379	TRUE
3-5	58	34.379	TRUE
4-5	2	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	52	34.379	TRUE
1-3	60	34.379	TRUE
1-4	99	34.379	TRUE
1-5	69	34.379	TRUE
2-3	8	34.379	FALSE
2-4	47	34.379	TRUE
2-5	17	34.379	FALSE
3-4	39	34.379	TRUE
3-5	9	34.379	FALSE
4-5	30	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	112	34.379	TRUE
1-3	70	34.379	TRUE
1-4	61	34.379	TRUE
1-5	57	34.379	TRUE
2-3	42	34.379	TRUE
2-4	51	34.379	TRUE
2-5	55	34.379	TRUE
3-4	9	34.379	FALSE
3-5	13	34.379	FALSE
4-5	4	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	11.5	34.379	FALSE
1-3	51.0	34.379	TRUE
1-4	53.5	34.379	TRUE
1-5	18.5	34.379	FALSE
2-3	62.5	34.379	TRUE
2-4	42.0	34.379	TRUE
2-5	7.0	34.379	FALSE
3-4	104.5	34.379	TRUE
3-5	69.5	34.379	TRUE
4-5	35.0	34.379	TRUE

**DTLZ2 - 20 Objetivos**

GDP

p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	42	34.379	TRUE
1-3	48	34.379	TRUE
1-4	114	34.379	TRUE
1-5	96	34.379	TRUE
2-3	6	34.379	FALSE
2-4	72	34.379	TRUE
2-5	54	34.379	TRUE
3-4	66	34.379	TRUE
3-5	48	34.379	TRUE
4-5	18	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	67	34.379	TRUE
1-3	53	34.379	TRUE
1-4	78	34.379	TRUE

1-5	87	34.379	TRUE
2-3	14	34.379	FALSE
2-4	11	34.379	FALSE
2-5	20	34.379	FALSE
3-4	25	34.379	FALSE
3-5	34	34.379	FALSE
4-5	9	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	114	34.379	TRUE
1-3	78	34.379	TRUE
1-4	67	34.379	TRUE
1-5	41	34.379	TRUE
2-3	36	34.379	TRUE
2-4	47	34.379	TRUE
2-5	73	34.379	TRUE
3-4	11	34.379	FALSE
3-5	37	34.379	TRUE
4-5	26	34.379	FALSE

**DTLZ3 - 2 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	56	34.379	TRUE
1-3	34	34.379	FALSE
1-4	50	34.379	TRUE
1-5	35	34.379	TRUE
2-3	22	34.379	FALSE
2-4	106	34.379	TRUE
2-5	91	34.379	TRUE
3-4	84	34.379	TRUE
3-5	69	34.379	TRUE
4-5	15	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	72	34.379	TRUE
1-3	40	34.379	TRUE
1-4	24	34.379	FALSE
1-5	13	34.379	FALSE
2-3	32	34.379	FALSE
2-4	96	34.379	TRUE
2-5	85	34.379	TRUE
3-4	64	34.379	TRUE
3-5	53	34.379	TRUE
4-5	11	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	62	34.379	TRUE
1-3	32	34.379	FALSE
1-4	46	34.379	TRUE
1-5	33	34.379	FALSE
2-3	30	34.379	FALSE
2-4	108	34.379	TRUE
2-5	95	34.379	TRUE
3-4	78	34.379	TRUE
3-5	65	34.379	TRUE
4-5	13	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	58	34.379	TRUE
1-3	31	34.379	FALSE
1-4	49	34.379	TRUE
1-5	40	34.379	TRUE
2-3	27	34.379	FALSE
2-4	107	34.379	TRUE
2-5	98	34.379	TRUE
3-4	80	34.379	TRUE
3-5	71	34.379	TRUE
4-5	9	34.379	FALSE

**DTLZ3 - 3 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	53	34.379	TRUE
1-3	12	34.379	FALSE
1-4	39	34.379	TRUE
1-5	48	34.379	TRUE
2-3	65	34.379	TRUE
2-4	92	34.379	TRUE
2-5	5	34.379	FALSE
3-4	27	34.379	FALSE
3-5	60	34.379	TRUE
4-5	87	34.379	TRUE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	96	34.379	TRUE
1-3	30	34.379	FALSE
1-4	39	34.379	TRUE
1-5	60	34.379	TRUE
2-3	66	34.379	TRUE
2-4	57	34.379	TRUE
2-5	36	34.379	TRUE
3-4	9	34.379	FALSE
3-5	30	34.379	FALSE
4-5	21	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	60	34.379	TRUE
1-3	16	34.379	FALSE
1-4	10	34.379	FALSE
1-5	61	34.379	TRUE
2-3	76	34.379	TRUE
2-4	70	34.379	TRUE
2-5	1	34.379	FALSE
3-4	6	34.379	FALSE
3-5	77	34.379	TRUE
4-5	71	34.379	TRUE

Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	45	34.379	TRUE
1-3	10	34.379	FALSE
1-4	71	34.379	TRUE
1-5	39	34.379	TRUE
2-3	55	34.379	TRUE
2-4	116	34.379	TRUE
2-5	84	34.379	TRUE
3-4	61	34.379	TRUE
3-5	29	34.379	FALSE

4-5 32 34.379 FALSE

**DTLZ3 - 5 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	68	34.379	TRUE
1-3	8	34.379	FALSE
1-4	6	34.379	FALSE
1-5	16	34.379	FALSE
2-3	76	34.379	TRUE
2-4	74	34.379	TRUE
2-5	52	34.379	TRUE
3-4	2	34.379	FALSE
3-5	24	34.379	FALSE
4-5	22	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	78	34.379	TRUE
1-3	28	34.379	FALSE
1-4	29	34.379	FALSE
1-5	41	34.379	TRUE
2-3	106	34.379	TRUE
2-4	49	34.379	TRUE
2-5	37	34.379	TRUE
3-4	57	34.379	TRUE
3-5	69	34.379	TRUE
4-5	12	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	85.5	34.379	TRUE
1-3	2.0	34.379	FALSE
1-4	24.5	34.379	FALSE
1-5	44.5	34.379	TRUE
2-3	87.5	34.379	TRUE
2-4	61.0	34.379	TRUE
2-5	41.0	34.379	TRUE
3-4	26.5	34.379	FALSE
3-5	46.5	34.379	TRUE
4-5	20.0	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	32	34.379	FALSE
1-3	54	34.379	TRUE
1-4	69	34.379	TRUE
1-5	44	34.379	TRUE
2-3	86	34.379	TRUE
2-4	101	34.379	TRUE
2-5	76	34.379	TRUE
3-4	15	34.379	FALSE
3-5	10	34.379	FALSE
4-5	25	34.379	FALSE

**DTLZ3 - 10 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons	obs.dif	critical.dif	difference
1-2	82.5	34.379	TRUE
1-3	21.5	34.379	FALSE

1-4	10.0	34.379	FALSE
1-5	8.5	34.379	FALSE
2-3	61.0	34.379	TRUE
2-4	72.5	34.379	TRUE
2-5	74.0	34.379	TRUE
3-4	11.5	34.379	FALSE
3-5	13.0	34.379	FALSE
4-5	1.5	34.379	FALSE

**IGDp**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	88	34.379	TRUE
1-3	13	34.379	FALSE
1-4	59	34.379	TRUE
1-5	36	34.379	TRUE
2-3	101	34.379	TRUE
2-4	29	34.379	FALSE
2-5	52	34.379	TRUE
3-4	72	34.379	TRUE
3-5	49	34.379	TRUE
4-5	23	34.379	FALSE

**Spacing**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	95	34.379	TRUE
1-3	25	34.379	FALSE
1-4	25	34.379	FALSE
1-5	35	34.379	TRUE
2-3	70	34.379	TRUE
2-4	70	34.379	TRUE
2-5	60	34.379	TRUE
3-4	0	34.379	FALSE
3-5	10	34.379	FALSE
4-5	10	34.379	FALSE

**Hypervolume**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	34	34.379	FALSE
1-3	60	34.379	TRUE
1-4	43	34.379	TRUE
1-5	61	34.379	TRUE
2-3	94	34.379	TRUE
2-4	77	34.379	TRUE
2-5	95	34.379	TRUE
3-4	17	34.379	FALSE
3-5	1	34.379	FALSE
4-5	18	34.379	FALSE

**DTLZ3 - 15 Objetivos****GDP**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	74.5	34.379	TRUE
1-3	26.0	34.379	FALSE
1-4	7.5	34.379	FALSE
1-5	15.5	34.379	FALSE
2-3	48.5	34.379	TRUE
2-4	82.0	34.379	TRUE
2-5	90.0	34.379	TRUE
3-4	33.5	34.379	FALSE
3-5	41.5	34.379	TRUE
4-5	8.0	34.379	FALSE

**IGDp**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	85	34.379	TRUE
1-3	19	34.379	FALSE
1-4	46	34.379	TRUE
1-5	33	34.379	FALSE
2-3	104	34.379	TRUE
2-4	39	34.379	TRUE
2-5	52	34.379	TRUE
3-4	65	34.379	TRUE
3-5	52	34.379	TRUE
4-5	13	34.379	FALSE

**Spacing**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	81.5	34.379	TRUE
1-3	22.5	34.379	FALSE
1-4	3.5	34.379	FALSE
1-5	7.0	34.379	FALSE
2-3	59.0	34.379	TRUE
2-4	85.0	34.379	TRUE
2-5	74.5	34.379	TRUE
3-4	26.0	34.379	FALSE
3-5	15.5	34.379	FALSE
4-5	10.5	34.379	FALSE

**Hypervolume**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	40	34.379	TRUE
1-3	50	34.379	TRUE
1-4	38	34.379	TRUE
1-5	52	34.379	TRUE
2-3	90	34.379	TRUE
2-4	78	34.379	TRUE
2-5	92	34.379	TRUE
3-4	12	34.379	FALSE
3-5	2	34.379	FALSE
4-5	14	34.379	FALSE

**DTLZ3 - 20 Objetivos****GDP**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	83.0	34.379	TRUE
1-3	19.5	34.379	FALSE
1-4	5.0	34.379	FALSE
1-5	12.5	34.379	FALSE
2-3	63.5	34.379	TRUE
2-4	78.0	34.379	TRUE
2-5	70.5	34.379	TRUE
3-4	14.5	34.379	FALSE
3-5	7.0	34.379	FALSE
4-5	7.5	34.379	FALSE

**IGDp**

Multiple comparisons between groups after Friedman test

p.value: 0.05

**Comparisons**

	obs.dif	critical.dif	difference
1-2	85	34.379	TRUE
1-3	19	34.379	FALSE
1-4	40	34.379	TRUE
1-5	34	34.379	FALSE
2-3	104	34.379	TRUE
2-4	45	34.379	TRUE
2-5	51	34.379	TRUE
3-4	59	34.379	TRUE
3-5	53	34.379	TRUE

4-5	6	34.379	FALSE
Spacing			
Multiple comparisons between groups after Friedman test			
p.value: 0.05			
Comparisons			
	obs.dif	critical.dif	difference
1-2	86	34.379	TRUE
1-3	18	34.379	FALSE
1-4	8	34.379	FALSE
1-5	23	34.379	FALSE
2-3	68	34.379	TRUE
2-4	78	34.379	TRUE
2-5	63	34.379	TRUE
3-4	10	34.379	FALSE
3-5	5	34.379	FALSE
4-5	15	34.379	FALSE

**DTLZ4 - 2 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	34	34.379	FALSE
1-3	16	34.379	FALSE
1-4	64	34.379	TRUE
1-5	48	34.379	TRUE
2-3	50	34.379	TRUE
2-4	30	34.379	FALSE
2-5	14	34.379	FALSE
3-4	80	34.379	TRUE
3-5	64	34.379	TRUE
4-5	16	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	49	34.379	TRUE
1-3	41	34.379	TRUE
1-4	74	34.379	TRUE
1-5	1	34.379	FALSE
2-3	8	34.379	FALSE
2-4	25	34.379	FALSE
2-5	48	34.379	TRUE
3-4	33	34.379	FALSE
3-5	40	34.379	TRUE
4-5	73	34.379	TRUE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	22	34.379	FALSE
1-3	38	34.379	TRUE
1-4	71	34.379	TRUE
1-5	74	34.379	TRUE
2-3	16	34.379	FALSE
2-4	49	34.379	TRUE
2-5	52	34.379	TRUE
3-4	33	34.379	FALSE
3-5	36	34.379	TRUE
4-5	3	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	49	34.379	TRUE
1-3	6	34.379	FALSE
1-4	49	34.379	TRUE
1-5	19	34.379	FALSE

2-3	43	34.379	TRUE
2-4	0	34.379	FALSE
2-5	68	34.379	TRUE
3-4	43	34.379	TRUE
3-5	25	34.379	FALSE
4-5	68	34.379	TRUE

**DTLZ4 - 3 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	57	34.379	TRUE
1-3	51	34.379	TRUE
1-4	108	34.379	TRUE
1-5	79	34.379	TRUE
2-3	6	34.379	FALSE
2-4	51	34.379	TRUE
2-5	22	34.379	FALSE
3-4	57	34.379	TRUE
3-5	28	34.379	FALSE
4-5	29	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	25	34.379	FALSE
1-3	2	34.379	FALSE
1-4	64	34.379	TRUE
1-5	69	34.379	TRUE
2-3	23	34.379	FALSE
2-4	39	34.379	TRUE
2-5	44	34.379	TRUE
3-4	62	34.379	TRUE
3-5	67	34.379	TRUE
4-5	5	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	89	34.379	TRUE
1-3	64	34.379	TRUE
1-4	67	34.379	TRUE
1-5	80	34.379	TRUE
2-3	25	34.379	FALSE
2-4	22	34.379	FALSE
2-5	9	34.379	FALSE
3-4	3	34.379	FALSE
3-5	16	34.379	FALSE
4-5	13	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	25	34.379	FALSE
1-3	43	34.379	TRUE
1-4	50	34.379	TRUE
1-5	67	34.379	TRUE
2-3	68	34.379	TRUE
2-4	75	34.379	TRUE
2-5	92	34.379	TRUE
3-4	7	34.379	FALSE
3-5	24	34.379	FALSE
4-5	17	34.379	FALSE

**DTLZ4 - 5 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05



Comparisons			
	obs.dif	critical.dif	difference
1-2	57	34.379	TRUE
1-3	33	34.379	FALSE
1-4	104	34.379	TRUE
1-5	106	34.379	TRUE
2-3	24	34.379	FALSE
2-4	47	34.379	TRUE
2-5	49	34.379	TRUE
3-4	71	34.379	TRUE
3-5	73	34.379	TRUE
4-5	2	34.379	FALSE

## IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	70	34.379	TRUE
1-3	25	34.379	FALSE
1-4	8	34.379	FALSE
1-5	7	34.379	FALSE
2-3	95	34.379	TRUE
2-4	62	34.379	TRUE
2-5	63	34.379	TRUE
3-4	33	34.379	FALSE
3-5	32	34.379	FALSE
4-5	1	34.379	FALSE

## Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	39.5	34.379	TRUE
1-3	62.5	34.379	TRUE
1-4	94.0	34.379	TRUE
1-5	99.0	34.379	TRUE
2-3	23.0	34.379	FALSE
2-4	54.5	34.379	TRUE
2-5	59.5	34.379	TRUE
3-4	31.5	34.379	FALSE
3-5	36.5	34.379	TRUE
4-5	5.0	34.379	FALSE

## Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	33	34.379	FALSE
1-3	50	34.379	TRUE
1-4	23	34.379	FALSE
1-5	85	34.379	TRUE
2-3	83	34.379	TRUE
2-4	56	34.379	TRUE
2-5	118	34.379	TRUE
3-4	27	34.379	FALSE
3-5	35	34.379	TRUE
4-5	62	34.379	TRUE

## DTLZ4 - 10 Objetivos

## GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	63	34.379	TRUE
1-3	31	34.379	FALSE
1-4	101	34.379	TRUE
1-5	100	34.379	TRUE
2-3	32	34.379	FALSE
2-4	38	34.379	TRUE
2-5	37	34.379	TRUE
3-4	70	34.379	TRUE
3-5	69	34.379	TRUE

4-5 1 34.379 FALSE

## IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	91	34.379	TRUE
1-3	6	34.379	FALSE
1-4	38	34.379	TRUE
1-5	42	34.379	TRUE
2-3	97	34.379	TRUE
2-4	53	34.379	TRUE
2-5	49	34.379	TRUE
3-4	44	34.379	TRUE
3-5	48	34.379	TRUE
4-5	4	34.379	FALSE

## Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	48	34.379	TRUE
1-3	51	34.379	TRUE
1-4	78	34.379	TRUE
1-5	93	34.379	TRUE
2-3	3	34.379	FALSE
2-4	30	34.379	FALSE
2-5	45	34.379	TRUE
3-4	27	34.379	FALSE
3-5	42	34.379	TRUE
4-5	15	34.379	FALSE

## Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	47	34.379	TRUE
1-3	63	34.379	TRUE
1-4	20	34.379	FALSE
1-5	29	34.379	FALSE
2-3	110	34.379	TRUE
2-4	67	34.379	TRUE
2-5	76	34.379	TRUE
3-4	43	34.379	TRUE
3-5	34	34.379	FALSE
4-5	9	34.379	FALSE

## DTLZ4 - 15 Objetivos

## GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	74	34.379	TRUE
1-3	31	34.379	FALSE
1-4	95	34.379	TRUE
1-5	100	34.379	TRUE
2-3	43	34.379	TRUE
2-4	21	34.379	FALSE
2-5	26	34.379	FALSE
3-4	64	34.379	TRUE
3-5	69	34.379	TRUE
4-5	5	34.379	FALSE

## IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons			
	obs.dif	critical.dif	difference
1-2	76	34.379	TRUE
1-3	9	34.379	FALSE
1-4	11	34.379	FALSE
1-5	22	34.379	FALSE
2-3	85	34.379	TRUE

2-4	65	34.379	TRUE
2-5	54	34.379	TRUE
3-4	20	34.379	FALSE
3-5	31	34.379	FALSE
4-5	11	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	61	34.379	TRUE
1-3	54	34.379	TRUE
1-4	82	34.379	TRUE
1-5	103	34.379	TRUE
2-3	7	34.379	FALSE
2-4	21	34.379	FALSE
2-5	42	34.379	TRUE
3-4	28	34.379	FALSE
3-5	49	34.379	TRUE
4-5	21	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	40	34.379	TRUE
1-3	61	34.379	TRUE
1-4	29	34.379	FALSE
1-5	40	34.379	TRUE
2-3	101	34.379	TRUE
2-4	69	34.379	TRUE
2-5	80	34.379	TRUE
3-4	32	34.379	FALSE
3-5	21	34.379	FALSE
4-5	11	34.379	FALSE

**DTLZ4 - 20 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	81	34.379	TRUE
1-3	28	34.379	FALSE
1-4	92	34.379	TRUE
1-5	89	34.379	TRUE
2-3	53	34.379	TRUE
2-4	11	34.379	FALSE
2-5	8	34.379	FALSE
3-4	64	34.379	TRUE
3-5	61	34.379	TRUE
4-5	3	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	70	34.379	TRUE
1-3	17	34.379	FALSE
1-4	7	34.379	FALSE
1-5	10	34.379	FALSE
2-3	87	34.379	TRUE
2-4	63	34.379	TRUE
2-5	60	34.379	TRUE
3-4	24	34.379	FALSE
3-5	27	34.379	FALSE
4-5	3	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	66	34.379	TRUE

1-3	56	34.379	TRUE
1-4	75	34.379	TRUE
1-5	88	34.379	TRUE
2-3	10	34.379	FALSE
2-4	9	34.379	FALSE
2-5	22	34.379	FALSE
3-4	19	34.379	FALSE
3-5	32	34.379	FALSE
4-5	13	34.379	FALSE

**DTLZ5 - 2 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	42	34.379	TRUE
1-3	75	34.379	TRUE
1-4	26	34.379	FALSE
1-5	16	34.379	FALSE
2-3	117	34.379	TRUE
2-4	68	34.379	TRUE
2-5	58	34.379	TRUE
3-4	49	34.379	TRUE
3-5	59	34.379	TRUE
4-5	10	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	30	34.379	FALSE
1-3	79	34.379	TRUE
1-4	103	34.379	TRUE
1-5	63	34.379	TRUE
2-3	49	34.379	TRUE
2-4	73	34.379	TRUE
2-5	33	34.379	FALSE
3-4	24	34.379	FALSE
3-5	16	34.379	FALSE
4-5	40	34.379	TRUE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	64	34.379	TRUE
1-3	56	34.379	TRUE
1-4	18	34.379	FALSE
1-5	30	34.379	FALSE
2-3	120	34.379	TRUE
2-4	82	34.379	TRUE
2-5	34	34.379	FALSE
3-4	38	34.379	TRUE
3-5	86	34.379	TRUE
4-5	48	34.379	TRUE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	44	34.379	TRUE
1-3	14	34.379	FALSE
1-4	80	34.379	TRUE
1-5	72	34.379	TRUE
2-3	30	34.379	FALSE
2-4	36	34.379	TRUE
2-5	28	34.379	FALSE
3-4	66	34.379	TRUE
3-5	58	34.379	TRUE
4-5	8	34.379	FALSE

**DTLZ5 - 3 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	35.0	34.379	TRUE
1-3	79.0	34.379	TRUE
1-4	15.5	34.379	FALSE
1-5	50.5	34.379	TRUE
2-3	114.0	34.379	TRUE
2-4	50.5	34.379	TRUE
2-5	85.5	34.379	TRUE
3-4	63.5	34.379	TRUE
3-5	28.5	34.379	FALSE
4-5	35.0	34.379	TRUE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	22	34.379	FALSE
1-3	35	34.379	TRUE
1-4	84	34.379	TRUE
1-5	94	34.379	TRUE
2-3	13	34.379	FALSE
2-4	62	34.379	TRUE
2-5	72	34.379	TRUE
3-4	49	34.379	TRUE
3-5	59	34.379	TRUE
4-5	10	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	72	34.379	TRUE
1-3	45	34.379	TRUE
1-4	14	34.379	FALSE
1-5	19	34.379	FALSE
2-3	117	34.379	TRUE
2-4	58	34.379	TRUE
2-5	53	34.379	TRUE
3-4	59	34.379	TRUE
3-5	64	34.379	TRUE
4-5	5	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	61	34.379	TRUE
1-3	11	34.379	FALSE
1-4	71	34.379	TRUE
1-5	34	34.379	FALSE
2-3	72	34.379	TRUE
2-4	10	34.379	FALSE
2-5	27	34.379	FALSE
3-4	82	34.379	TRUE
3-5	45	34.379	TRUE
4-5	37	34.379	TRUE

**DTLZ5 - 5 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	47	34.379	TRUE
1-3	67	34.379	TRUE
1-4	4	34.379	FALSE
1-5	26	34.379	FALSE
2-3	114	34.379	TRUE

2-4	51	34.379	TRUE
2-5	73	34.379	TRUE
3-4	63	34.379	TRUE
3-5	41	34.379	TRUE
4-5	22	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	13	34.379	FALSE
1-3	2	34.379	FALSE
1-4	78	34.379	TRUE
1-5	63	34.379	TRUE
2-3	15	34.379	FALSE
2-4	91	34.379	TRUE
2-5	76	34.379	TRUE
3-4	76	34.379	TRUE
3-5	61	34.379	TRUE
4-5	15	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	73.0	34.379	TRUE
1-3	24.0	34.379	FALSE
1-4	56.5	34.379	TRUE
1-5	59.5	34.379	TRUE
2-3	97.0	34.379	TRUE
2-4	16.5	34.379	FALSE
2-5	13.5	34.379	FALSE
3-4	80.5	34.379	TRUE
3-5	83.5	34.379	TRUE
4-5	3.0	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	13	34.379	FALSE
1-3	34	34.379	FALSE
1-4	73	34.379	TRUE
1-5	28	34.379	FALSE
2-3	47	34.379	TRUE
2-4	60	34.379	TRUE
2-5	15	34.379	FALSE
3-4	107	34.379	TRUE
3-5	62	34.379	TRUE
4-5	45	34.379	TRUE

**DTLZ5 - 10 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	61.0	34.379	TRUE
1-3	51.0	34.379	TRUE
1-4	1.5	34.379	FALSE
1-5	8.5	34.379	FALSE
2-3	112.0	34.379	TRUE
2-4	59.5	34.379	TRUE
2-5	52.5	34.379	TRUE
3-4	52.5	34.379	TRUE
3-5	59.5	34.379	TRUE
4-5	7.0	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	16	34.379	FALSE

1-3	24	34.379	FALSE
1-4	65	34.379	TRUE
1-5	62	34.379	TRUE
2-3	40	34.379	TRUE
2-4	81	34.379	TRUE
2-5	78	34.379	TRUE
3-4	41	34.379	TRUE
3-5	38	34.379	TRUE
4-5	3	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	44.5	34.379	TRUE
1-3	11.0	34.379	FALSE
1-4	79.5	34.379	TRUE
1-5	79.5	34.379	TRUE
2-3	55.5	34.379	TRUE
2-4	35.0	34.379	TRUE
2-5	35.0	34.379	TRUE
3-4	90.5	34.379	TRUE
3-5	90.5	34.379	TRUE
4-5	0.0	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	15	34.379	FALSE
1-3	24	34.379	FALSE
1-4	72	34.379	TRUE
1-5	62	34.379	TRUE
2-3	39	34.379	TRUE
2-4	57	34.379	TRUE
2-5	47	34.379	TRUE
3-4	96	34.379	TRUE
3-5	86	34.379	TRUE
4-5	10	34.379	FALSE

**DTLZ5 - 15 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	55	34.379	TRUE
1-3	51	34.379	TRUE
1-4	24	34.379	FALSE
1-5	2	34.379	FALSE
2-3	106	34.379	TRUE
2-4	31	34.379	FALSE
2-5	53	34.379	TRUE
3-4	75	34.379	TRUE
3-5	53	34.379	TRUE
4-5	22	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	9	34.379	FALSE
1-3	52	34.379	TRUE
1-4	73	34.379	TRUE
1-5	59	34.379	TRUE
2-3	61	34.379	TRUE
2-4	82	34.379	TRUE
2-5	68	34.379	TRUE
3-4	21	34.379	FALSE
3-5	7	34.379	FALSE
4-5	14	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	41	34.379	TRUE
1-3	8	34.379	FALSE
1-4	89	34.379	TRUE
1-5	83	34.379	TRUE
2-3	49	34.379	TRUE
2-4	48	34.379	TRUE
2-5	42	34.379	TRUE
3-4	97	34.379	TRUE
3-5	91	34.379	TRUE
4-5	6	34.379	FALSE

**DTLZ5 - 20 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	67	34.379	TRUE
1-3	49	34.379	TRUE
1-4	2	34.379	FALSE
1-5	19	34.379	FALSE
2-3	116	34.379	TRUE
2-4	69	34.379	TRUE
2-5	48	34.379	TRUE
3-4	47	34.379	TRUE
3-5	68	34.379	TRUE
4-5	21	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	30	34.379	FALSE
1-3	38	34.379	TRUE
1-4	44	34.379	TRUE
1-5	53	34.379	TRUE
2-3	68	34.379	TRUE
2-4	74	34.379	TRUE
2-5	83	34.379	TRUE
3-4	6	34.379	FALSE
3-5	15	34.379	FALSE
4-5	9	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	57	34.379	TRUE
1-3	16	34.379	FALSE
1-4	47	34.379	TRUE
1-5	92	34.379	TRUE
2-3	73	34.379	TRUE
2-4	10	34.379	FALSE
2-5	35	34.379	TRUE
3-4	63	34.379	TRUE
3-5	108	34.379	TRUE
4-5	45	34.379	TRUE

**DTLZ6 - 2 Objetivos**

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	117	34.379	TRUE
1-3	40	34.379	TRUE
1-4	41	34.379	TRUE
1-5	87	34.379	TRUE
2-3	77	34.379	TRUE

2-4	76	34.379	TRUE
2-5	30	34.379	FALSE
3-4	1	34.379	FALSE
3-5	47	34.379	TRUE
4-5	46	34.379	TRUE

**IGDp**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	108	34.379	TRUE
1-3	66	34.379	TRUE
1-4	6	34.379	FALSE
1-5	60	34.379	TRUE
2-3	42	34.379	TRUE
2-4	102	34.379	TRUE
2-5	48	34.379	TRUE
3-4	60	34.379	TRUE
3-5	6	34.379	FALSE
4-5	54	34.379	TRUE

**Spacing**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	72	34.379	TRUE
1-3	31	34.379	FALSE
1-4	22	34.379	FALSE
1-5	72	34.379	TRUE
2-3	103	34.379	TRUE
2-4	50	34.379	TRUE
2-5	0	34.379	FALSE
3-4	53	34.379	TRUE
3-5	103	34.379	TRUE
4-5	50	34.379	TRUE

**Hypervolume**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	90.0	48.61925	TRUE
1-3	85.5	48.61925	TRUE
1-4	9.0	48.61925	FALSE
1-5	38.5	48.61925	FALSE
2-3	4.5	48.61925	FALSE
2-4	99.0	48.61925	TRUE
2-5	51.5	48.61925	TRUE
3-4	94.5	48.61925	TRUE
3-5	47.0	48.61925	FALSE
4-5	47.5	48.61925	FALSE

**DTLZ6 - 3 Objetivos****GDP**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	87	34.379	TRUE
1-3	24	34.379	FALSE
1-4	33	34.379	FALSE
1-5	69	34.379	TRUE
2-3	111	34.379	TRUE
2-4	54	34.379	TRUE
2-5	18	34.379	FALSE
3-4	57	34.379	TRUE
3-5	93	34.379	TRUE
4-5	36	34.379	TRUE

**IGDp**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
--	---------	--------------	------------

1-2	106	34.379	TRUE
1-3	33	34.379	FALSE
1-4	30	34.379	FALSE
1-5	61	34.379	TRUE
2-3	73	34.379	TRUE
2-4	76	34.379	TRUE
2-5	45	34.379	TRUE
3-4	3	34.379	FALSE
3-5	28	34.379	FALSE
4-5	31	34.379	FALSE

**Spacing**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	70	34.379	TRUE
1-3	21	34.379	FALSE
1-4	33	34.379	FALSE
1-5	72	34.379	TRUE
2-3	49	34.379	TRUE
2-4	103	34.379	TRUE
2-5	2	34.379	FALSE
3-4	54	34.379	TRUE
3-5	51	34.379	TRUE
4-5	105	34.379	TRUE

**Hypervolume**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	95	48.61925	TRUE
1-3	42	48.61925	FALSE
1-4	37	48.61925	FALSE
1-5	66	48.61925	TRUE
2-3	53	48.61925	TRUE
2-4	58	48.61925	TRUE
2-5	29	48.61925	FALSE
3-4	5	48.61925	FALSE
3-5	24	48.61925	FALSE
4-5	29	48.61925	FALSE

**DTLZ6 - 5 Objetivos****GDP**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	109	34.379	TRUE
1-3	8	34.379	FALSE
1-4	62	34.379	TRUE
1-5	66	34.379	TRUE
2-3	101	34.379	TRUE
2-4	47	34.379	TRUE
2-5	43	34.379	TRUE
3-4	54	34.379	TRUE
3-5	58	34.379	TRUE
4-5	4	34.379	FALSE

**IGDp**

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	99	34.379	TRUE
1-3	12	34.379	FALSE
1-4	55	34.379	TRUE
1-5	53	34.379	TRUE
2-3	111	34.379	TRUE
2-4	44	34.379	TRUE
2-5	46	34.379	TRUE
3-4	67	34.379	TRUE
3-5	65	34.379	TRUE
4-5	2	34.379	FALSE

Spacing  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	119	34.379	TRUE
1-3	43	34.379	TRUE
1-4	53	34.379	TRUE
1-5	80	34.379	TRUE
2-3	76	34.379	TRUE
2-4	66	34.379	TRUE
2-5	39	34.379	TRUE
3-4	10	34.379	FALSE
3-5	37	34.379	TRUE
4-5	27	34.379	FALSE

Hypervolume  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	40	48.61925	FALSE
1-3	66	48.61925	TRUE
1-4	34	48.61925	FALSE
1-5	65	48.61925	TRUE
2-3	106	48.61925	TRUE
2-4	74	48.61925	TRUE
2-5	105	48.61925	TRUE
3-4	32	48.61925	FALSE
3-5	1	48.61925	FALSE
4-5	31	48.61925	FALSE

#### DTLZ6 - 10 Objetivos

GDP  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	108.5	34.379	TRUE
1-3	7.0	34.379	FALSE
1-4	62.5	34.379	TRUE
1-5	64.5	34.379	TRUE
2-3	101.5	34.379	TRUE
2-4	46.0	34.379	TRUE
2-5	44.0	34.379	TRUE
3-4	55.5	34.379	TRUE
3-5	57.5	34.379	TRUE
4-5	2.0	34.379	FALSE

IGDp  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	97	34.379	TRUE
1-3	16	34.379	FALSE
1-4	52	34.379	TRUE
1-5	52	34.379	TRUE
2-3	113	34.379	TRUE
2-4	45	34.379	TRUE
2-5	45	34.379	TRUE
3-4	68	34.379	TRUE
3-5	68	34.379	TRUE
4-5	0	34.379	FALSE

Spacing  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	116	34.379	TRUE
1-3	29	34.379	FALSE
1-4	60	34.379	TRUE
1-5	90	34.379	TRUE
2-3	87	34.379	TRUE
2-4	56	34.379	TRUE

2-5	26	34.379	FALSE
3-4	31	34.379	FALSE
3-5	61	34.379	TRUE
4-5	30	34.379	FALSE

Hypervolume  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	17	48.61925	FALSE
1-3	15	48.61925	FALSE
1-4	1	48.61925	FALSE
1-5	18	48.61925	FALSE
2-3	32	48.61925	FALSE
2-4	16	48.61925	FALSE
2-5	35	48.61925	FALSE
3-4	16	48.61925	FALSE
3-5	3	48.61925	FALSE
4-5	19	48.61925	FALSE

#### DTLZ6 - 15 Objetivos

GDP  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	105	34.379	TRUE
1-3	0	34.379	FALSE
1-4	60	34.379	TRUE
1-5	60	34.379	TRUE
2-3	105	34.379	TRUE
2-4	45	34.379	TRUE
2-5	45	34.379	TRUE
3-4	60	34.379	TRUE
3-5	60	34.379	TRUE
4-5	0	34.379	FALSE

IGDp  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	101	34.379	TRUE
1-3	8	34.379	FALSE
1-4	53	34.379	TRUE
1-5	59	34.379	TRUE
2-3	109	34.379	TRUE
2-4	48	34.379	TRUE
2-5	42	34.379	TRUE
3-4	61	34.379	TRUE
3-5	67	34.379	TRUE
4-5	6	34.379	FALSE

Spacing  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	116	34.379	TRUE
1-3	22	34.379	FALSE
1-4	61	34.379	TRUE
1-5	81	34.379	TRUE
2-3	94	34.379	TRUE
2-4	55	34.379	TRUE
2-5	35	34.379	TRUE
3-4	39	34.379	TRUE
3-5	59	34.379	TRUE
4-5	20	34.379	FALSE

Hypervolume  
Multiple comparisons between groups after Friedman test  
p.value: 0.05  
Comparisons

	obs.dif	critical.dif	difference
1-2	24	48.61925	FALSE

1-3	12	48.61925	FALSE
1-4	15	48.61925	FALSE
1-5	21	48.61925	FALSE
2-3	12	48.61925	FALSE
2-4	9	48.61925	FALSE
2-5	45	48.61925	FALSE
3-4	3	48.61925	FALSE
3-5	33	48.61925	FALSE
4-5	36	48.61925	FALSE

**DTLZ6 - 20 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	103.5	34.379	TRUE
1-3	3.0	34.379	FALSE
1-4	61.5	34.379	TRUE
1-5	55.5	34.379	TRUE
2-3	106.5	34.379	TRUE
2-4	42.0	34.379	TRUE
2-5	48.0	34.379	TRUE
3-4	64.5	34.379	TRUE
3-5	58.5	34.379	TRUE
4-5	6.0	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	101	34.379	TRUE
1-3	7	34.379	FALSE
1-4	56	34.379	TRUE
1-5	55	34.379	TRUE
2-3	108	34.379	TRUE
2-4	45	34.379	TRUE
2-5	46	34.379	TRUE
3-4	63	34.379	TRUE
3-5	62	34.379	TRUE
4-5	1	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	92	34.379	TRUE
1-3	20	34.379	FALSE
1-4	99	34.379	TRUE
1-5	64	34.379	TRUE
2-3	72	34.379	TRUE
2-4	7	34.379	FALSE
2-5	28	34.379	FALSE
3-4	79	34.379	TRUE
3-5	44	34.379	TRUE
4-5	35	34.379	TRUE

**DTLZ7 - 2 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	116	34.379	TRUE
1-3	22	34.379	FALSE
1-4	86	34.379	TRUE
1-5	56	34.379	TRUE
2-3	94	34.379	TRUE
2-4	30	34.379	FALSE
2-5	60	34.379	TRUE
3-4	64	34.379	TRUE
3-5	34	34.379	FALSE
4-5	30	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	115	34.379	TRUE
1-3	26	34.379	FALSE
1-4	89	34.379	TRUE
1-5	60	34.379	TRUE
2-3	89	34.379	TRUE
2-4	26	34.379	FALSE
2-5	55	34.379	TRUE
3-4	63	34.379	TRUE
3-5	34	34.379	FALSE
4-5	29	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	109.5	34.379	TRUE
1-3	18.0	34.379	FALSE
1-4	67.5	34.379	TRUE
1-5	75.0	34.379	TRUE
2-3	91.5	34.379	TRUE
2-4	42.0	34.379	TRUE
2-5	34.5	34.379	TRUE
3-4	49.5	34.379	TRUE
3-5	57.0	34.379	TRUE
4-5	7.5	34.379	FALSE

Hypervolume

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	108.0	48.61925	TRUE
1-3	11.0	48.61925	FALSE
1-4	70.5	48.61925	TRUE
1-5	60.0	48.61925	TRUE
2-3	119.0	48.61925	TRUE
2-4	37.5	48.61925	FALSE
2-5	48.0	48.61925	FALSE
3-4	81.5	48.61925	TRUE
3-5	71.0	48.61925	TRUE
4-5	10.5	48.61925	FALSE

**DTLZ7 - 3 Objetivos**

GDP

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	108	34.379	TRUE
1-3	6	34.379	FALSE
1-4	75	34.379	TRUE
1-5	51	34.379	TRUE
2-3	102	34.379	TRUE
2-4	33	34.379	FALSE
2-5	57	34.379	TRUE
3-4	69	34.379	TRUE
3-5	45	34.379	TRUE
4-5	24	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test  
p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	90	34.379	TRUE
1-3	30	34.379	FALSE
1-4	52	34.379	TRUE
1-5	38	34.379	TRUE

2-3	120	34.379	TRUE
2-4	38	34.379	TRUE
2-5	52	34.379	TRUE
3-4	82	34.379	TRUE
3-5	68	34.379	TRUE
4-5	14	34.379	FALSE

## Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	85	34.379	TRUE
1-3	28	34.379	FALSE
1-4	32	34.379	FALSE
1-5	66	34.379	TRUE
2-3	113	34.379	TRUE
2-4	53	34.379	TRUE
2-5	19	34.379	FALSE
3-4	60	34.379	TRUE
3-5	94	34.379	TRUE
4-5	34	34.379	FALSE

## Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	86.5	48.61925	TRUE
1-3	95.0	48.61925	TRUE
1-4	42.5	48.61925	FALSE
1-5	71.0	48.61925	TRUE
2-3	181.5	48.61925	TRUE
2-4	44.0	48.61925	FALSE
2-5	15.5	48.61925	FALSE
3-4	137.5	48.61925	TRUE
3-5	166.0	48.61925	TRUE
4-5	28.5	48.61925	FALSE

**DTLZ7 - 5 Objetivos**

## GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	120.0	34.379	TRUE
1-3	30.0	34.379	FALSE
1-4	83.5	34.379	TRUE
1-5	66.5	34.379	TRUE
2-3	90.0	34.379	TRUE
2-4	36.5	34.379	TRUE
2-5	53.5	34.379	TRUE
3-4	53.5	34.379	TRUE
3-5	36.5	34.379	TRUE
4-5	17.0	34.379	FALSE

## IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	90	34.379	TRUE
1-3	30	34.379	FALSE
1-4	47	34.379	TRUE
1-5	43	34.379	TRUE
2-3	120	34.379	TRUE
2-4	43	34.379	TRUE
2-5	47	34.379	TRUE
3-4	77	34.379	TRUE
3-5	73	34.379	TRUE
4-5	4	34.379	FALSE

## Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
--	---------	--------------	------------

1-2	64	34.379	TRUE
1-3	46	34.379	TRUE
1-4	2	34.379	FALSE
1-5	54	34.379	TRUE
2-3	110	34.379	TRUE
2-4	66	34.379	TRUE
2-5	10	34.379	FALSE
3-4	44	34.379	TRUE
3-5	100	34.379	TRUE
4-5	56	34.379	TRUE

## Hypervolume

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	53	34.379	TRUE
1-3	30	34.379	FALSE
1-4	53	34.379	TRUE
1-5	74	34.379	TRUE
2-3	83	34.379	TRUE
2-4	0	34.379	FALSE
2-5	21	34.379	FALSE
3-4	83	34.379	TRUE
3-5	104	34.379	TRUE
4-5	21	34.379	FALSE

**DTLZ7 - 10 Objetivos**

## GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	120.0	34.379	TRUE
1-3	30.0	34.379	FALSE
1-4	79.5	34.379	TRUE
1-5	70.5	34.379	TRUE
2-3	90.0	34.379	TRUE
2-4	40.5	34.379	TRUE
2-5	49.5	34.379	TRUE
3-4	49.5	34.379	TRUE
3-5	40.5	34.379	TRUE
4-5	9.0	34.379	FALSE

## IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	90	34.379	TRUE
1-3	28	34.379	FALSE
1-4	41	34.379	TRUE
1-5	52	34.379	TRUE
2-3	118	34.379	TRUE
2-4	49	34.379	TRUE
2-5	38	34.379	TRUE
3-4	69	34.379	TRUE
3-5	80	34.379	TRUE
4-5	11	34.379	FALSE

## Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

## Comparisons

	obs.dif	critical.dif	difference
1-2	43.0	34.379	TRUE
1-3	44.5	34.379	TRUE
1-4	12.0	34.379	FALSE
1-5	67.0	34.379	TRUE
2-3	87.5	34.379	TRUE
2-4	31.0	34.379	FALSE
2-5	24.0	34.379	FALSE
3-4	56.5	34.379	TRUE
3-5	111.5	34.379	TRUE
4-5	55.0	34.379	TRUE



Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	13	34.379	FALSE
1-3	40	34.379	TRUE
1-4	54	34.379	TRUE
1-5	73	34.379	TRUE
2-3	53	34.379	TRUE
2-4	41	34.379	TRUE
2-5	60	34.379	TRUE
3-4	94	34.379	TRUE
3-5	113	34.379	TRUE
4-5	19	34.379	FALSE

#### DTLZ7 - 15 Objetivos

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	120	34.379	TRUE
1-3	30	34.379	FALSE
1-4	78	34.379	TRUE
1-5	72	34.379	TRUE
2-3	90	34.379	TRUE
2-4	42	34.379	TRUE
2-5	48	34.379	TRUE
3-4	48	34.379	TRUE
3-5	42	34.379	TRUE
4-5	6	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	89	34.379	TRUE
1-3	30	34.379	FALSE
1-4	38	34.379	TRUE
1-5	53	34.379	TRUE
2-3	119	34.379	TRUE
2-4	51	34.379	TRUE
2-5	36	34.379	TRUE
3-4	68	34.379	TRUE
3-5	83	34.379	TRUE
4-5	15	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	27.5	34.379	FALSE
1-3	47.5	34.379	TRUE
1-4	15.0	34.379	FALSE

1-5	67.5	34.379	TRUE
2-3	75.0	34.379	TRUE
2-4	12.5	34.379	FALSE
2-5	40.0	34.379	TRUE
3-4	62.5	34.379	TRUE
3-5	115.0	34.379	TRUE
4-5	52.5	34.379	TRUE

#### DTLZ7 - 20 Objetivos

GDP

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	120	34.379	TRUE
1-3	30	34.379	FALSE
1-4	86	34.379	TRUE
1-5	64	34.379	TRUE
2-3	90	34.379	TRUE
2-4	34	34.379	FALSE
2-5	56	34.379	TRUE
3-4	56	34.379	TRUE
3-5	34	34.379	FALSE
4-5	22	34.379	FALSE

IGDp

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	89	34.379	TRUE
1-3	28	34.379	FALSE
1-4	35	34.379	TRUE
1-5	59	34.379	TRUE
2-3	117	34.379	TRUE
2-4	54	34.379	TRUE
2-5	30	34.379	FALSE
3-4	63	34.379	TRUE
3-5	87	34.379	TRUE
4-5	24	34.379	FALSE

Spacing

Multiple comparisons between groups after Friedman test

p.value: 0.05

Comparisons

	obs.dif	critical.dif	difference
1-2	30.5	34.379	FALSE
1-3	41.0	34.379	TRUE
1-4	34.0	34.379	FALSE
1-5	71.5	34.379	TRUE
2-3	71.5	34.379	TRUE
2-4	3.5	34.379	FALSE
2-5	41.0	34.379	TRUE
3-4	75.0	34.379	TRUE
3-5	112.5	34.379	TRUE
4-5	37.5	34.379	TRUE

## ANEXO A

### BENCHMARKS UTILIZADOS

#### A.1 Funções de *benchmarks* com um critério

a)  $F_1$ : *Shifted Sphere Function*

$$F_1(x) = \sum_{i=1}^D z_i^2 + f\_bias_1, \quad (A.1)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$D = 100$  dimensões.  $\mathbf{o} = [o_1, o_2, \dots, o_D]$ : Ótimo Global.

**Propriedades:** unimodal, shifted (deslocada), separável, escalável.

$$x \in [-100, 100]^D,$$

$$\text{Ótimo Global: } x^* = \mathbf{o}, F_1(x^*) = f\_bias_1 = -450$$

b)  $F_2$ : *Schwefel's Problem 2.21*

$$F_2(x) = \max_{i=1} \{|z_i|, 1 \leq i \leq D\} + f\_bias_2, \quad (A.2)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$D = 100$  dimensões.  $\mathbf{o} = [o_1, o_2, \dots, o_D]$ : Ótimo Global.

**Propriedades:** unimodal, shifted, não-separável, escalável.

$$x \in [-100, 100]^D,$$

$$\text{Ótimo Global: } x^* = \mathbf{o}, F_2(x^*) = f\_bias_2 = -450$$

c)  $F_3$ : *Shifted Rosenbrocks's Function*

$$F_3(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f\_bias_3, \quad (A.3)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$D = 100$  dimensões.  $\mathbf{o} = [o_1, o_2, \dots, o_D]$ : Ótimo Global.

**Propriedades:** multi-modal, shifted (deslocada), não-separável, escalável.

$$x \in [-100, 100]^D,$$

$$\text{Ótimo Global: } x^* = \mathbf{o}, F_3(x^*) = f\_bias_3 = 390$$

**d)  $F_4$ : Shifted Rastrigin's Function**

$$F_4(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f\_bias_4, \quad (A.4)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$D = 100$  dimensões.  $\mathbf{o} = [o_1, o_2, \dots, o_D]$ : Ótimo Global.

**Propriedades:** multi-modal, shifted (deslocada), separável, escalável.

$$x \in [-5, 5]^D,$$

$$\text{Ótimo Global: } x^* = \mathbf{o}, F_4(x^*) = f\_bias_4 = -330$$

**e)  $F_5$ : Shifted Griewank's Function**

$$F_5(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + f\_bias_5, \quad (A.5)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$D = 100$  dimensões.  $\mathbf{o} = [o_1, o_2, \dots, o_D]$ : Ótimo Global.

**Propriedades:** multi-modal, shifted (deslocada), não-separável, escalável.

$$x \in [-600, 600]^D,$$

$$\text{Ótimo Global: } x^* = \mathbf{o}, F_5(x^*) = f\_bias_5 = -180$$

**f)  $F_6$ : Shifted Ackley's Function**

$$F_6(x) = -20 \exp(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e + f\_bias_6, \quad (A.6)$$

$$\mathbf{z} = \mathbf{x} - \mathbf{o}, \quad \mathbf{x} = [x_1, x_2, \dots, x_D]$$

$D = 100$  dimensões.  $\mathbf{o} = [o_1, o_2, \dots, o_D]$ : Ótimo Global.

**Propriedades:** multi-modal, shifted (deslocada), separável, escalável.

$$x \in [-32, 32]^D,$$

$$\text{Ótimo Global: } x^* = \mathbf{o}, F_6(x^*) = f\_bias_6 = -140$$

Todas as funções são definidas como problemas de minimização.

## A.2 Funções de *benchmarks* com mais de um critério

### a) Problema DTLZ 1

A função de teste DTLZ1 é um problema M-objetivo com uma fronteira de Pareto linear. Este problema é definido como:

$$\min f(\vec{X})$$

$$\begin{cases} f_1(\vec{X}) = \frac{1}{2}x_1x_2\dots x_{M-1}(1 + g(\vec{X}_M)), \\ f_2(\vec{X}) = \frac{1}{2}x_1x_2\dots(1 - x_{M-1})(1 + g(\vec{X}_M)), \\ \dots \\ f_{M-1}(\vec{X}) = \frac{1}{2}x_1(1 - x_2)(1 + g(\vec{X}_M)), \\ f_M(\vec{X}) = \frac{1}{2}(1 - x_1)(1 + g(\vec{X}_M)), \\ \text{onde } g(\vec{X}_M) = 100 \left( |\vec{X}_M| + \sum_{x_i \in \vec{X}_M} (x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right), \\ \text{sujeito a } 0 \leq x_i \leq 1, \forall_i = 1, 2, \dots, n. \end{cases} \quad (\text{A.7})$$

A fronteira de Pareto ótima corresponde a  $x_i = 0,5$  (para todo  $x_i \in \vec{X}_M$ ) e os valores das funções objetivo ficam no hiperplano linear:  $\sum_{m=1}^M (f_m^*) = 0,5$ . A dificuldade deste problema está relacionada à convergência para este hiperplano. O espaço de busca contém  $(11^k - 1)$  fronteiras de Pareto locais (sub-ótimas). MOEAs podem ser atraídos à estas regiões antes de atingir a fronteira de Pareto real. A Figura A.1(a) ilustra a fronteira de Pareto real para a função DTLZ1 considerando três objetivos.

### b) Problema DTLZ2

Esta função é utilizada para investigar a capacidade de escalar de um MOEA em termos de número de objetivos. É um problema adequado para avaliar a habilidade de convergência dos algoritmos. O problema DTLZ2 é definido por:

$$\min f(\vec{X})$$

$$\begin{cases} f_1(\vec{X}) = (1 + g(\vec{X}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \dots \cos(x_{M-1} \cdot \frac{\pi}{2}), \\ f_2(\vec{X}) = (1 + g(\vec{X}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \dots \sin(x_{M-1} \cdot \frac{\pi}{2}), \\ \dots \\ f_M(\vec{X}) = (1 + g(\vec{X}_M)) \sin(x_1 \cdot \frac{\pi}{2}), \\ \text{onde } g(\vec{X}_M) = \sum_{x_i \in \vec{X}_M} (x_i - 0,5)^2, \\ \text{sujeito a } 0 \leq x_i \leq 1, \forall_i = 1, 2, \dots, n. \end{cases} \quad (\text{A.8})$$

As soluções de Pareto ótimas correspondem a  $x_i = 0,5$  (para todo  $x_i \in \vec{X}_M$ ) e os valores das funções objetivo devem satisfazer  $\sum_{m=1}^M (f_m^*)^2 = 1$ . A Figura A.1(b) ilustra uma fronteira de Pareto real obtida pela função DTLZ2 considerando três objetivos.

### c) Problema DTLZ3

Este problema é uma combinação dos dois problemas anteriores. Os autores utilizaram a base da função DTLZ2 e substituíram a função  $g$  deste *benchmark* pela função  $g$  do problema DTLZ1. Este novo problema foi proposto para investigar a habilidade do MOEA em convergir para a fronteira de Pareto real e é definido por:

$$\min f(\vec{X})$$

$$\begin{cases} f_1(\vec{X}) = (1 + g(\vec{X}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \dots \cos(x_{M-1} \cdot \frac{\pi}{2}), \\ f_2(\vec{X}) = (1 + g(\vec{X}_M)) \cos(x_1 \cdot \frac{\pi}{2}) \dots \sin(x_{M-1} \cdot \frac{\pi}{2}), \\ \dots \\ f_M(\vec{X}) = (1 + g(\vec{X}_M)) \sin(x_1 \cdot \frac{\pi}{2}), \\ \text{onde } g(\vec{X}_M) = 100 \left( |\vec{X}_M| + \sum_{x_i \in \vec{X}_M} (x_i - 0,5)^2 - \cos(20\pi(x_i - 0,5)) \right), \\ \text{sujeito a } 0 \leq x_i \leq 1, \forall_i = 1, 2, \dots, n. \end{cases} \quad (\text{A.9})$$

A Figura A.1(c) ilustra uma fronteira de Pareto real obtida pela função DTLZ3 para três objetivos.

#### d) Problema DTLZ4

Este problema é uma modificação do problema DTLZ2 e é usado para investigar a habilidade dos MOEAs em manter uma boa distribuição das soluções. A modificação consiste no mapeamento  $x_i \rightarrow x_i^\alpha$ , onde  $\alpha = 100$ . O problema DTLZ4 é definido por:

$$\begin{cases} f_1(\vec{X}) = (1 + g(\vec{X}_M)) \cos(x_1^\alpha \cdot \frac{\pi}{2}) \dots \cos(x_{M-1}^\alpha \cdot \frac{\pi}{2}), \\ f_2(\vec{X}) = (1 + g(\vec{X}_M)) \cos(x_1^\alpha \cdot \frac{\pi}{2}) \dots \sin(x_{M-1}^\alpha \cdot \frac{\pi}{2}), \\ \dots \\ f_M(\vec{X}) = (1 + g(\vec{X}_M)) \sin(x_1^\alpha \cdot \frac{\pi}{2}), \\ \text{onde } g(\vec{X}_M) = \sum_{x_i \in \vec{X}_M} (x_i - 0,5)^2, \\ \text{sujeito a } 0 \leq x_i \leq 1, \forall_i = 1, 2, \dots, n. \end{cases} \quad (\text{A.10})$$

A Figura A.1(d) ilustra a fronteira de Pareto real obtida pela função DTLZ4 com três objetivos. Este problema tende a gerar pontos em uma região específica da fronteira de Pareto. Especificamente no plano  $f_M - f_1$ . Considerando que os MOEAs tentam encontrar um conjunto de soluções bem distribuído em cada execução, este problema pode impedir que tais algoritmos obtenham um bom desempenho.

#### e) Problema DTLZ5

Neste problema, a função DTLZ2 é utilizada como base e as funções trigonométricas são mapeadas em função de  $\theta_i$ . Onde  $\theta_i = \frac{\pi}{4(1+g(\vec{X}_M))} \cdot (1 + 2g(\vec{X}_M)x_i)$ , para  $i = 2, 3, \dots, (M-1)$ . Nesse problema a fronteira de Pareto é definida por apenas uma curva e não por toda a região de um quadrante de esfera. Este problema é definido por:

$$\min f(\vec{X})$$

$$\left\{ \begin{array}{l} f_1(\vec{X}) = (1 + g(\vec{X}_M)) \cos(\theta_1 \cdot \frac{\pi}{2}) \dots \cos(\theta_{M-1} \cdot \frac{\pi}{2}), \\ f_2(\vec{X}) = (1 + g(\vec{X}_M)) \cos(\theta_1 \cdot \frac{\pi}{2}) \dots \sin(\theta_{M-1} \cdot \frac{\pi}{2}), \\ \dots \\ f_M(\vec{X}) = (1 + g(\vec{X}_M)) \sin(\theta_1 \cdot \frac{\pi}{2}), \\ \text{com } \theta_i = \frac{\pi}{4(1+g(\vec{X}_M))} \cdot (1 + 2g(\vec{X}_M)x_i), \forall_i = 2, 3, \dots, (M-1), \\ \text{onde } g(\vec{X}_M) = \sum_{x_i \in \vec{X}_M} (x_i - 0,5)^2, \\ \text{sujeito a } 0 \leq x_i \leq 1, \forall_i = 1, 2, \dots, n. \end{array} \right. \quad (\text{A.11})$$

A fronteira de Pareto real corresponde a  $x_i = 0,5$  (para todo  $x_i \in \vec{X}_M$ ) e os valores das funções objetivo satisfazem  $\sum_{m=1}^M (f_m^*)^2 = 1$ . A Figura A.1(e) ilustra uma fronteira de Pareto real obtida pela função DTLZ5 para três objetivos.

#### f) Problema DTLZ6

Aqui, a mesma ideia utilizada para criar o problema DTLZ3 foi reiterada: a função  $g$  do problema DTLZ5 é modificada, gerando o problema DTLZ6. Este problema também foi proposto para investigar a escalabilidade dos algoritmos em termos de convergência, à medida que o número de objetivos aumenta e é definido por:

$$\min f(\vec{X})$$

$$\left\{ \begin{array}{l} f_1(\vec{X}) = (1 + g(\vec{X}_M)) \cos(\theta_1 \cdot \frac{\pi}{2}) \dots \cos(\theta_{M-1} \cdot \frac{\pi}{2}), \\ f_2(\vec{X}) = (1 + g(\vec{X}_M)) \cos(\theta_1 \cdot \frac{\pi}{2}) \dots \sin(\theta_{M-1} \cdot \frac{\pi}{2}), \\ \dots \\ f_M(\vec{X}) = (1 + g(\vec{X}_M)) \sin(\theta_1 \cdot \frac{\pi}{2}), \\ \text{com } \theta_i = \frac{\pi}{4(1+g(\vec{X}_M))} \cdot (1 + 2g(\vec{X}_M)x_i), \forall_i = 2, 3, \dots, (M-1), \\ \text{onde } g(\vec{X}_M) = \sum_{x_i \in \vec{X}_M} (x_i)^{0,1}, \\ \text{sujeito a } 0 \leq x_i \leq 1, \forall_i = 1, 2, \dots, n. \end{array} \right. \quad (\text{A.12})$$

Devido a maior complexidade da função  $g$ , o problema DTLZ6 é considerado mais difícil que o *benchmark* DTLZ5, do qual é derivado. Aqui, a fronteira de Pareto real corresponde a  $x_i = 0$  (para todo  $x_i \in \vec{X}_M$ ). A Figura A.1(f) ilustra a fronteira de Pareto real obtida pela função DTLZ6 para três objetivos.

#### g) Problema DTLZ7

Neste problema existem  $2^{M-1}$  regiões desconexas da fronteira de Pareto no espaço de busca. Este problema testa a habilidade de um algoritmo em manter subpopulações em diferentes regiões da fronteira real e é definido por:

$$\min f(\vec{X})$$

$$\begin{cases}
f_1(\vec{X}) = x_1, \\
f_2(\vec{X}) = x_2, \\
\vdots \\
f_{M-1}(\vec{X}) = x_{M-1}, \\
f_M(\vec{X}) = (1 + g(\vec{X}_M))h(f_1, f_2, \dots, f_{M-1}, g), \\
\text{onde } g(\vec{X}_M) = 1 + \frac{9}{|\vec{X}_M|} \sum_{x_i \in \vec{X}_M} (x_i), \\
\text{e } h(f_1, f_2, \dots, f_{M-1}, g) = M - \sum_{i=1}^{M-1} \left[ \frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right], \\
\text{sujeito a } 0 \leq x_i \leq 1, \forall i = 1, 2, \dots, n.
\end{cases} \quad (\text{A.13})$$

A Figura A.1(g) ilustra a fronteira de Pareto real obtida pela função DTLZ7 para três objetivos.

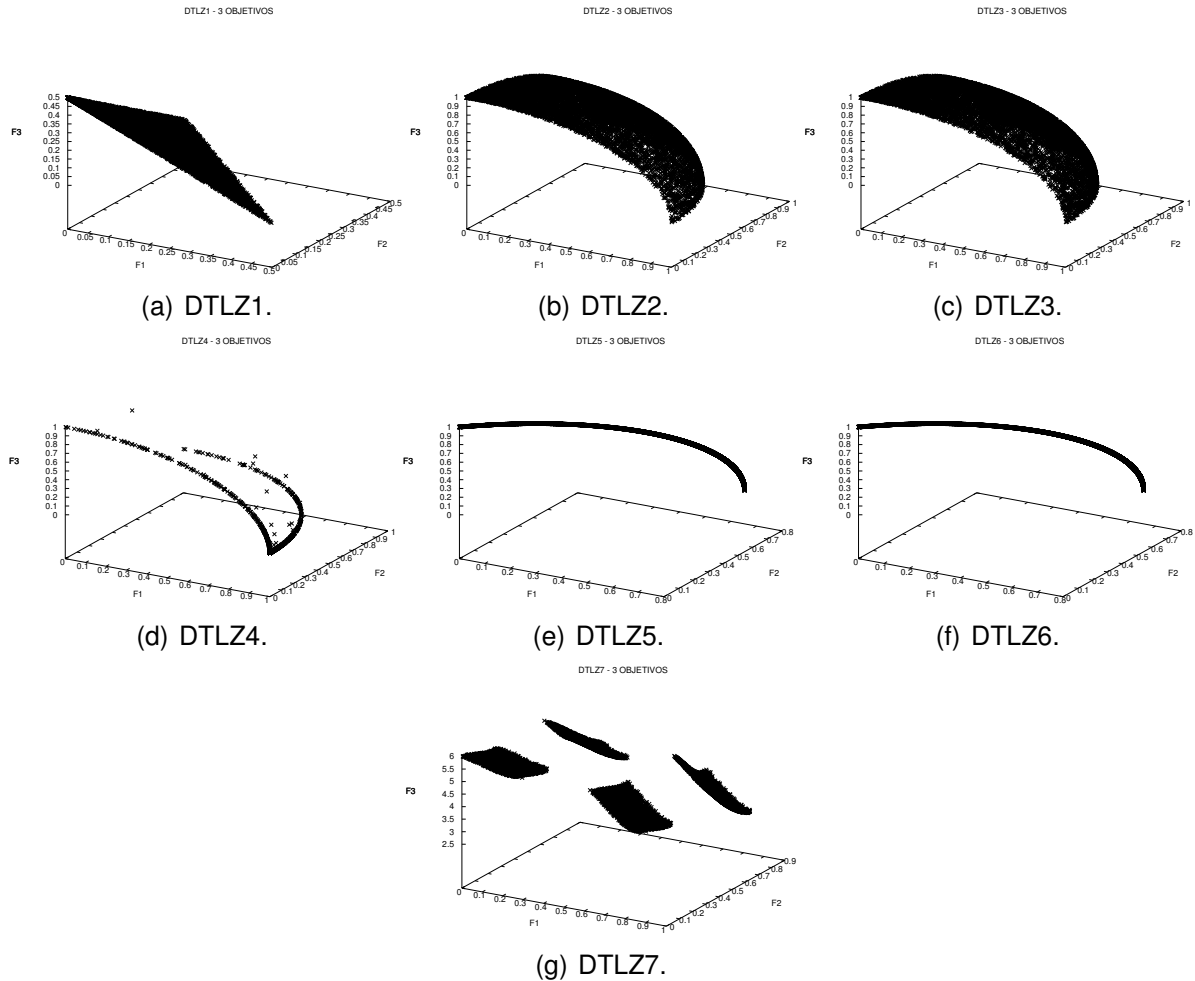


Figura A.1: Fronteiras de Pareto real relativas ao conjunto de problemas DTLZ para três objetivos.